

5 Cada vez somos más

Revista de Software Libre ATIX 2009

Equipo Editorial

Colaboradores

Dirección y Coordinación General

Esteban Saavedra López (jesaavedra@opentelematics.org)

Diseño y Maquetación

Jenny Saavedra López (jennysaavedra@gmail.com)

Esteban Saavedra López (jesaavedra@opentelematics.org)

Revisiones

Esteban Saavedra López

Jenny Saavedra López

Noticias

Ivonne Menacho

Jenny Saavedra López

Marcia Velasquez

Autores Frecuentes

Arnold Guzmán

Carlos Ramos

Ernesto Rico Smith

Esteban Saavedra López

Joseph Sandoval

Rocio Figueroa

Rafael Rendón

Herramientas

La edición de esta revista fue realizada de forma integra haciendo uso de Software Libre









Palabra quechua,

con un sentimiento profundo

y con gran significado filosófico

El que lo sabe

El que lo intenta

El que lo puede

El que lo logra

Editorial

Si hacemos una retrospectiva de los inicios del movimiento de software libre, veremos que en ese entonces fueron muy pocos los interesados en este tema y más pocos aún los hozados en continuar durante los siguientes años; pero los que iniciaron este movimiento deben sentirse orgullosos de ver que hoy en día somos millones las personas inmersas en el mundo del software libre, donde día a día se suman más adeptos entre usuarios, simpatizantes y activistas, que promueven la colaboración, la ética y la libertad sobre todas las cosas

Otro motivo de sentirse orgulloso es la presencia del software libre en escuelas, universidades y empresas en general ha crecido notablemente, dando lugar al nacimiento de numerosos proyectos en diversas áreas, los cuales van contribuyendo a procesos de automatización, educación e interacción entre personas y empresas.

Cada vez somos más, un título que refleja de forma clara cuanto ha crecido el número de usuarios de software libre en todo el mundo: “cada vez somos más”, no solo es un término estático, sino más bien es un término lleno de dinamismo que muestra el aumento de personas, usuarios, activistas y proyectos inmersos en el mundo del software libre.

En éste décimo primer número ponderamos los proyectos latinos finalistas del Comunity Choice Awards, seguros estamos que estos abren las puertas para otros proyectos de origen latino, los mejores deseos de éxito para todos ellos.

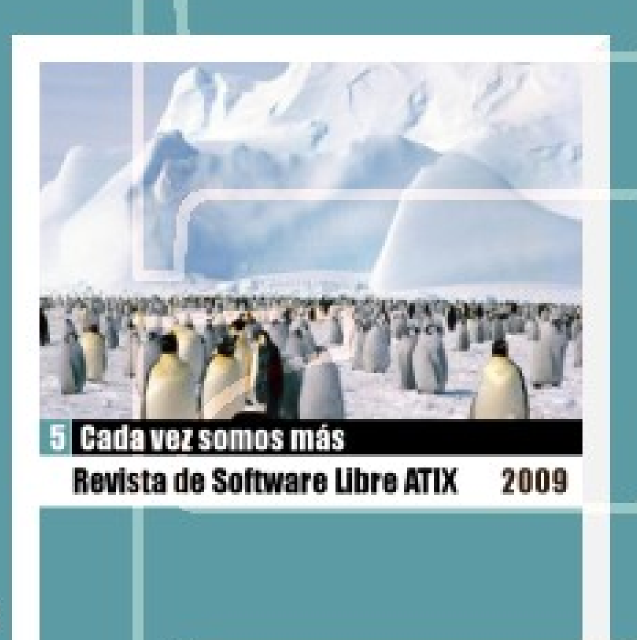
Cada vez somos más y así nos fortalecemos

Bienvenidos a nuestro décimo primer número

Esteban Saavedra López
Director y Coordinador General

Contenido

Liberado el 13 de julio de 2009



Plataforma móvil Android

Hoy día la tecnología móvil está invadiendo poco a poco todos los espacios en cuanto a comunicaciones y entretenimiento se refiere. Prepararnos para usarlas es vital para todo usuario de estas tecnologías de la información y comunicación.

Introducción

Hace ya varios años cuando IBM lanzó al mercado el primer smartphone bautizado con el nombre de Símon en el año 1992, año en el que la historia de los teléfonos inteligentes comienza marcando la diferencia entre la telefonía tradicional y la nueva forma de comunicación. Esta nueva modalidad de dispositivos de comunicación propone ser más que un simple teléfono al incluir manejo de agenda, contactos, correo electrónico, acceso a Internet; utilidades tales como reloj mundial, calculadoras, entre otras, siendo las mencionadas las características más comunes entre los teléfonos inteligentes. A partir de ese momento la tendencia en materia de telefonía comenzó a cambiar...Hasta lo que es hoy.



Gráfico 1. Dispositivos móviles.

Aproximadamente 2 décadas después hoy podemos encontrar varias opciones en cuanto a teléfonos inteligentes se refiere, por ejemplo, Blackberry, Symbian, Windows Mobile, iPhone, propuestas basadas en linux como Mobilinux, Convergent Linux y más recientemente la plataforma android también partiendo de Linux. Este último es sobre el cual se enfoca este artículo. Cabe decir que cada una de estas opciones mencionadas son plataformas, sistemas operativos para dispositivos móviles, por tanto al igual que las computadoras no están necesariamente ligadas a un dispositivo en específico, con algunas excepciones como el iPhone por ejemplo.

Android

Android es un sistema operativo, una plataforma de software para dispositivos móviles, está basado en el sistema operativo GNU/Linux y por ende es una plataforma libre y que además comparte las características de los sistemas GNU/Linux, por ejemplo, sistema multiusuario, multitarea (como casi todo SO moderno) y demás características propias de este sistema.



Gráfico 2. Logo Android.

Android nació como un proyecto desarrollado en un principio por la empresa de servicios de Internet muy conocida, Google. Ya para cuando se presentó la plataforma al público Android contaba con el apoyo de varias compañías de la industria informática, la

fundación Open Handset Alliance, aproximadamente 48 compañías que apoyan y fomentan la creación de estándares abiertos.



Gráfico 3. SO android en acción.

A diferencia de otras plataformas tales como iPhone, la cual por razones de seguridad,argumenta la empresa, no permite directamente la creación de aplicaciones por parte de terceros, (aunque recientemente las cosas han cambiado ya que Apple liberó un SDK para iPhone), esta nueva plataforma sí permite el desarrollo de aplicaciones por parte de terceros (característica natural de las plataformas abiertas). Es decir, cualquier persona con los conocimientos suficientes tendrá la oportunidad de crear aplicaciones para sus dispositivos o para redistribución, para tal caso, android proporciona un SDK gratuito para el desarrollo de aplicaciones.



Gráfico 4. Google Android.

Características

A continuación se listan algunas de las características más representativas de esta plataforma, enfocadas a ventajas y desventajas.

Ventajas

- ✔ Plataforma libre
- ✔ Permite creación de aplicaciones por parte de terceros
- ✔ Prioridad equitativa entre aplicaciones nativas del sistema y aplicaciones de terceros.
- ✔ Creación de aplicaciones rápida y fácil
- ✔ Proporciona SDK

Desventajas

- ✔ Se ha criticado que Android no es un Sistema totalmente Abierto y que Google quiera controlar el sistema siendo que alegan ser un sistema totalmente abierto.



Gráfico 5. Características de Android.

Expectativas para Android

Hoy día Android es una propuesta joven y prometedora que llega al mercado sembrando diversidad de opiniones con respecto al porvenir que le depara a éste nuevo SO para móviles. En un mercado de sistemas operativos para móviles, según datos proporcionados por Gartner para el cuarto trimestre del 2008, Symbian se

posiciona a la cabeza con un 47.1% de presencia en el mercado, seguido por BlacBerry OS con un 19.5%, precediéndole Windows Mobile con un 12.4%, iPhone OS alcanzando ya un 10.7% y los SO basados en Linux con apenas 8.4%, de antemano se pronostica una batalla muy dura (y prolongada) para Android si quiere figurar en estas estadísticas.

Table 3
Worldwide: Smartphone Sales to End Users by Operating System, 4Q08 (Thousands of Units)

Company	4Q08 Sales	Market Share 4Q08 (%)	4Q07 Sales	Market Share 4Q07 (%)	Growth 4Q07-4Q08 (%)
Symbian	17,949.1	47.1	22,902.5	62.3	-21.6
Research In Motion	7,442.6	19.5	4,024.7	10.9	84.9
Microsoft Windows Mobile	4,713.9	12.4	4,374.4	11.9	7.8
Mac OS X	4,079.4	10.7	1,928.3	5.2	111.6
Linux	3,194.9	8.4	2,675.9	7.3	19.4
Palm OS	326.5	0.9	449.1	1.2	-27.3
Other OSs	436.9	1.1	411.3	1.1	6.2
Total	**38,143.3**	**100.0**	**36,766.1**	**100.0**	**3.7**

Gráfico 6. SO para móviles y su participación en el mercado según Gartner, 2008.

Por la parte de las empresas, que se han mostrado escépticas ante este acontecimiento, HTC, una empresa Asiática dedicada a las tecnologías móviles, fue la primera que se "atrevió" utilizar Android con SO para sus productos(actualmente oferta el HTC Dream y HTC Magic), marcando la pauta para otras empresas, tales como Samsung y LG que tienen ya en sus planes la inclusión de este nuevo SO para algunos de sus productos.



Gráfico 7. Primeros móviles con Android.

En el otro lado del mercado, la parte de los usuarios, las reacciones son similares. El tema de Android a sido tema de conversación de un sin número de foros, blogs, páginas de noticias, etc. Muchos usuarios afines al software libre esperan ansiosos su primer dispositivo con este SO (me incluyo desde luego), con la esperanza y creencia de que que sus prestaciones tienen futuro en el mundo de los móviles, por otra parte existen también los que opinan lo contrario argumentando que Android tiene un ambiente gráfico “feo”, que su política de acceso equitativo al equipo tanto para aplicaciones nativas como para aplicaciones de terceros va ser su punto débil, que es lento, etc.

Ahora bien, desde mi muy subjetivo punto de vista, yo opino que debemos prepararnos porque nos espera Android para rato. Si bien la competencia es fuerte, Android trae todas las herramientas necesarias para hacer frente al mercado, incluso para los usuarios más escépticos. El hecho de carecer de restricciones es uno de los puntos que pienso serán el fuerte de este SO.



Gráfico 8. Emulador de android visitando ATIX en la web.

Desarrollo de aplicaciones en Android

Al igual que la mayoría de las plataformas de la competencia, Android, nos ofrece desde un principio un conjunto de herramientas para el desarrollo de software(SDK) que hace posible crear aplicaciones para este SO. Android pone a disposición de los usuarios y desarrolladores un SDK sin costo alguno en su página oficial, `www.android.com` . A la fecha de publicación de este documento el SDK se encuentra en su versión 1.5.



Gráfico 9. Android developer.

El lenguaje que se ha elegido para la creación de aplicaciones es JAVA, para el cual se ha creado su API correspondiente y también algunos plugings para los IDEs de JAVA más populares (bueno de hecho es solo para eclipse, para los demás se tienen que hacer adaptaciones). Cabe mencionar que las aplicaciones no utilizan la máquina virtual de JAVA desarrollada por Sun Microsystems, sino una máquina virtual desarrollada por Google para optimizar el código para Android, Dalvik.

En la página oficial se pude encontrar documentación y algunos ejemplos sencillos de aplicaciones para Android, así como el manual de referencia de las APIs, donde se detallan cada una de ellas, sus clases, métodos, interfaces, etc. El próximo artículo lo dedicare a la realización de un pequeño programa para esta plataforma, con lo cual espero quede más claro esto del desarrollo de aplicaciones para Android, de momento lo dejamos así.

Alternativas

Opciones para elegir son muchas, tanto de sistemas privativos (lamentablemente, los dominantes) como opciones de código abierto basadas en los sistemas Linux. Como en nuestro caso lo que nos interesa son las opciones encaminadas al software libre, a continuación les dejo algunas de las opciones alternativas con respecto a SO para móviles.



Gráfico 10. Linux para móviles.

Convergent Linux



Es un sistema operativo para móviles desarrollada por la empresa a la Mobile basado en Linux. Según sus creadores es el primer sistema que puede ser portado a varias plataformas, incluso si utilizan otros tipos de procesadores. Por naturaleza es código abierto y una buena opción alternativa a Android. Más información en `www.a-la-mobile.com`.

LiMo



Es también un proyecto joven desarrollado por varias empresas como Motorola, NTT, NEC, DoCoMo, Samsung entre otras, iniciado en enero de 2007. Un SO basado en Linux también diseñado para ser independiente del hardware de los dispositivos. Para más información en `www.limofoundation.org`.

Mobilinux



Otra alternativa libre para dispositivos móviles con Linux como sistema base, fue presentado en el año 2005 por la empresa Montavista. Este SO utiliza el kernel 2.6, un entorno gráfico basado en KDrive y tecnología GTK. Más sobre Mobilinux en `www.mvista.com`.

Qtopia



Esta es una propuesta por parte de Trolltech... si... la misma que es dueña de las bibliotecas QT. Presenta 2 versiones de SO, Opie, un sistema libre bajo licencia GPL y otra versión comercial, así también 2 ediciones, una para móviles y otra para PDAs. En **www.qtextended.org** pueden encontrar más información.

Otras

Existen otras alternativas como ARM Linux Mobile Platform, también Access Linux Platform (ALP) u OpenMoko, entre muchas otras. El hecho es que de que hay de donde... lo hay! Es cuestión de que se decidan xD.

Conclusión

Como pueden ver tenemos una muy buena propuesta en puerta, con buenas prestaciones, open source, con buenos patrocinadores, tiene todo lo necesario para un usuario y mucho más para los desarrolladores. Pues he aquí el dato, los usuarios son lo que tienen la última palabra.

Referencias

[1] "Smartphone", www.es.wikipedia.org.

[2] www.android.com

[3] Gartner, Inc. "Gartner Says Worldwide Smartphone Sales Reached Its Lowest Growth Rate With 3.7 Per Cent Increase in Fourth Quarter of 2008", www.gartner.com

Autor



Rafael Rendón Pablo
Estudiante de Ingeniería en Sistemas Computacionales
Usuario de software libre
smart.dyr@gmail.com
www.inforscience.260mb.com



Introducción al Proyecto Mono

Mono es el nombre de un proyecto de código abierto iniciado por Ximian y actualmente impulsado por Novell, para crear un grupo de herramientas libres, basadas en GNU/Linux y compatibles con .NET



Introducción

Mono es un proyecto de código abierto liberado por Novell para crear un conjunto de herramientas .NET compatibles con el estándar ECMA, incluyendo entre otros un compilador C# y un Lenguaje Común de Tiempo de Ejecución.

¿Qué es Mono?

El sitio oficial de Mono[1] define al proyecto literalmente como: "una iniciativa de desarrollo abierto patrocinada por Novell para desarrollar una versión UNIX de código abierto de la plataforma de desarrollo Microsoft .NET"; ésta es la plataforma de desarrollo creada por Microsoft para sus sistemas, que busca de forma uniforme crear un conjunto de componentes que permitan al programador mejorar significativamente su productividad, incluyendo una gran variedad de librerías para problemas computacionales comunes además de una máquina virtual que administra la ejecución de los programas escritos específicamente para la plataforma.

Mono es algo más que la implementación de Microsoft .NET para sistemas UNIX, pues a lo largo de su historia se han desarrollado componentes, librerías y plataformas donde se ha utilizado exclusivamente Mono para programarlas, por ejemplo:

- ✔ Gtk#: recubrimientos del toolkit de interfaz gráfica Gtk+ para sistemas UNIX y Microsoft Windows.
- ✔ Dbus-Sharp: implementación del protocolo DBus para lenguajes administrados.
- ✔ Mono.Addins: plataforma genérico para crear aplicaciones extensibles, y para crear librerías que extienden de otras aplicaciones.



Figura 1: Arquitectura de Mono simplificada

Todas estas librerías y componentes fueron desarrolladas en Mono porque ésta contiene los siguientes componentes para construir software:

- ✔ Una máquina virtual con una Infraestructura de Lenguaje Común (CLI por sus siglas en inglés) que contiene un cargador de clases, compilador al-momento o JIT además de un recolector de basura en tiempo de ejecución.
- ✔ Una librería de clases que puede trabajar con cualquier lenguaje que funcione sobre un Lenguaje Común en Tiempo de Ejecución (CLR por sus siglas en inglés). Ambas librerías, las provistas por aquellas compatibles

con clases .NET al igual que las provistas por Mono son incluidas.

- ✔ Un compilador para el lenguaje C#, aunque también existe el compilador para el lenguaje VB.NET, que no se incluye ahora por defecto pero también existe.

Plataforma Mono

Hoy en día las distribuciones de Linux más populares incluyen, o plantean incluir, por defecto la Plataforma Mono para permitir la ejecución de las aplicaciones existentes basadas en ésta. En caso que la distribución no incluya por defecto los paquetes necesarios para ejecutar la Plataforma Mono, siempre se pueden descargar de la página oficial[1]. La distribuciones que ahora incluyen Mono están concientes de que estas nuevas aplicaciones enriquecen los escritorios Linux, ya que proveen más alternativas para el usuario final, algunos ejemplos de estas aplicaciones son:

- ✔ **Tomboy[2]:** aplicación de escritorio para tomar notas en Linux y UNIX.
- ✔ **Banshee[3]:** herramienta de reproducción y organización de música.
- ✔ **Beagle[4]:** herramienta de búsqueda inteligente que encuentra lo que quieres.
- ✔ **F-Spot[5]:** aplicación de administración de fotos para el escritorio Gnome.
- ✔ **Gnome-Do[6]:** herramienta que permite realizar tareas comunes de forma simple y eficiente.

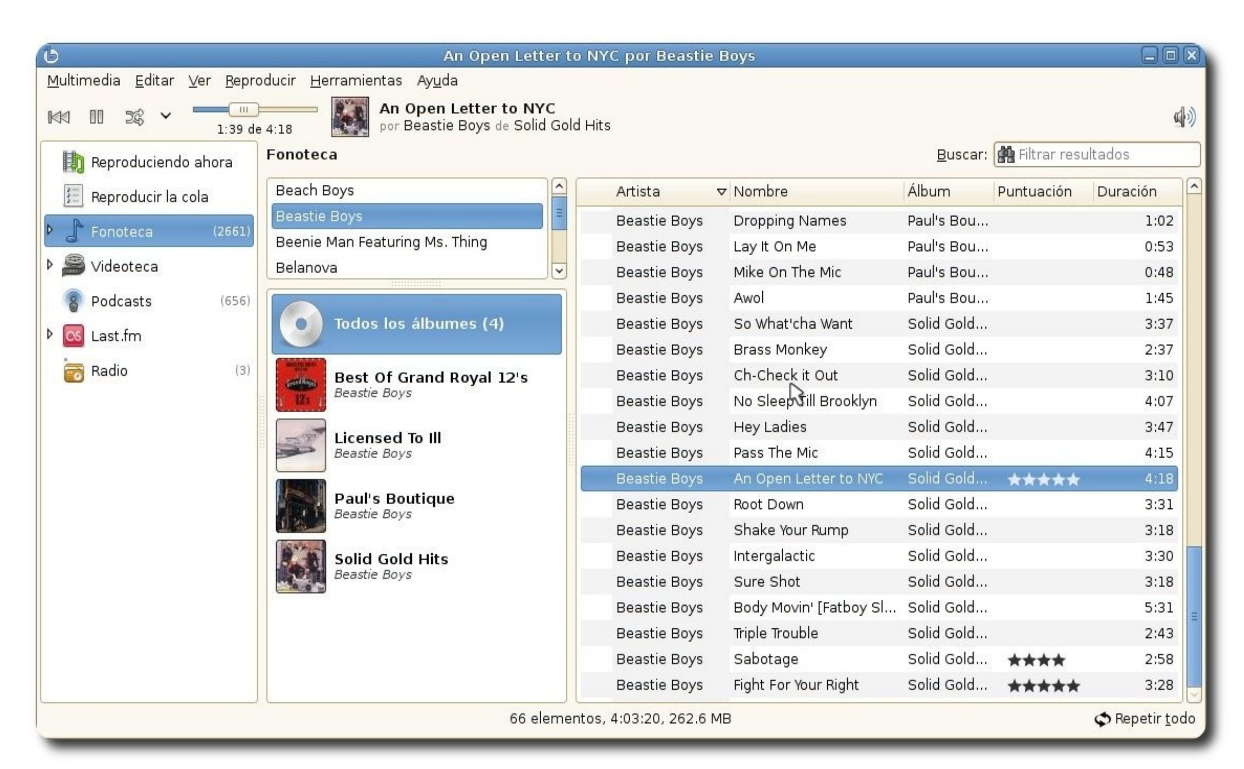


Figura 2: Reproductor Multimedia Banshee

Las aplicaciones listadas previamente fueron desarrolladas exclusivamente utilizando la plataforma Mono y Linux como base de desarrollo, ahora muchas de ellas ya funcionan en sistemas Microsoft Windows donde la plataforma Microsoft .NET se encuentra disponible; todo ésto gracias a que la plataforma Mono implementa los estándares ECMA[7] que definen el formato interno que las aplicaciones contienen y además el lenguaje que comúnmente se utiliza: C#.

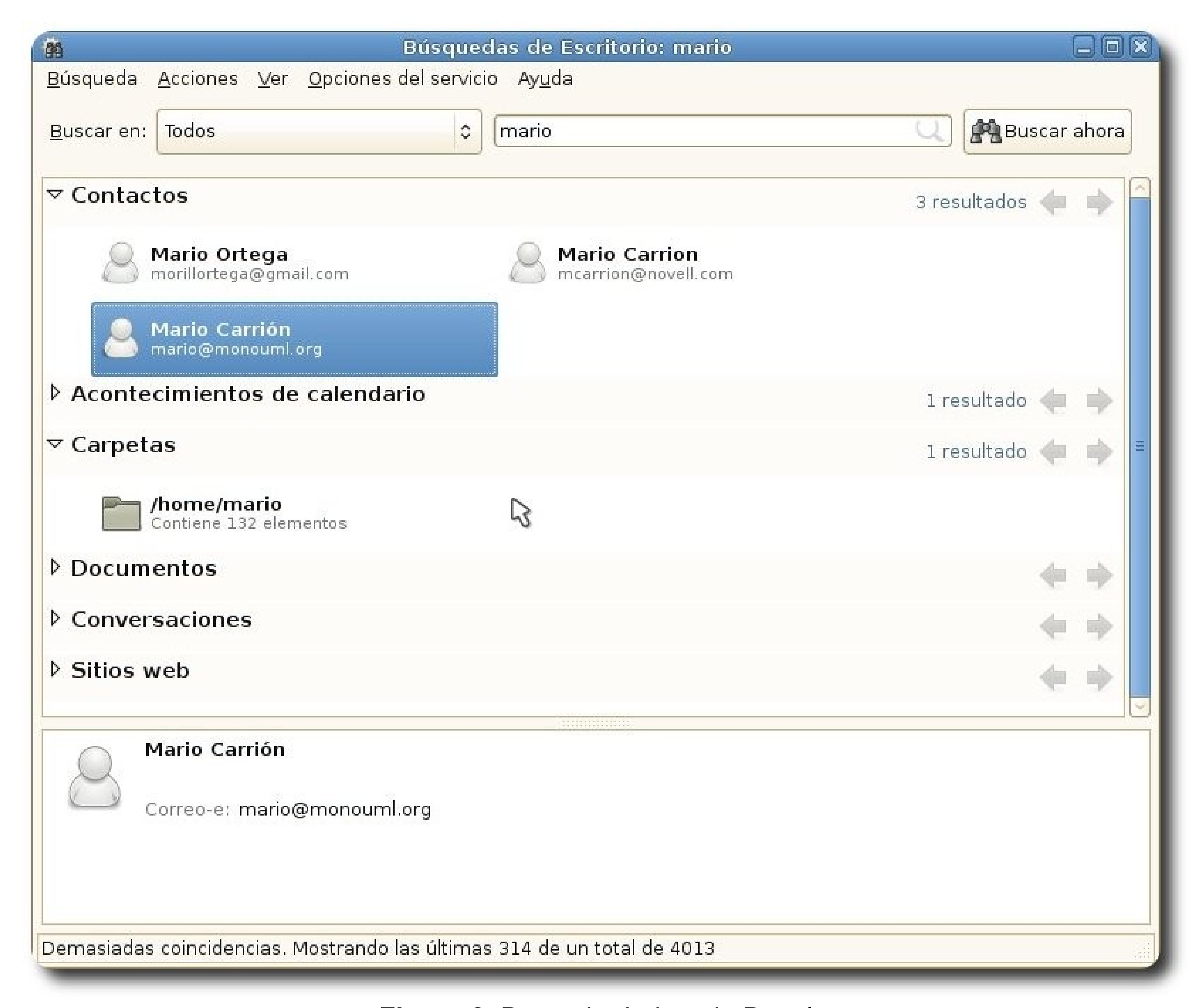


Figura 3: Buscador indexado Beagle

Beneficios al escoger Mono para desarrollo de aplicaciones

- ✔ **Popularidad**: Basado en el éxito de .NET, hay millones de desarrolladores que tienen la experiencia de construir aplicaciones en C#. También hay miles de libros, sitios web, tutoriales, y ejemplos de código para ayudar en cualquier problema imaginable.

- ✔ **Programación de alto nivel**: Todos los lenguajes de Mono se benefician de las características del tiempo de ejecución, como administración automática de memoria, reflexión, genéricos, e hilos. Estas características permiten al desarrollador concentrarse en escribir aplicaciones en vez de escribir código de infraestructura de sistema.

- ✔ **Librería de clase base**: El tener una librería de clases comprensiva provee miles de clases ya construidas para incrementar productividad. ¿Necesitas código para conexiones de red o tablas hash? No hay necesidad de escribir las propias, pues ya están incluidas dentro de la plataforma.

- ✔ **Multi-plataforma**: Mono está construido para ser multi-plataforma. Mono funciona sobre Linux, Microsoft Windows, Mac OS X, BSD, Sun Solaris, Nintendo Wii, Sony Playtation 3, Apple iPhone. Además se ejecuta sobre x86, x86-64, IA64, PowerPC, SPARC (32), ARM, Alpha, s390, s390x (32 y 64 bits) y más. Desarrollar tu aplicación con Mono permite ejecutarla prácticamente en cualquier computadora existente.

- ✔ **Lenguaje Común en Tiempo de Ejecución**, (CLR, por sus siglas en ingles): Permite escoger el lenguaje de programación que mejor funcione para tus objetivos, y éste puede interoperar con código escrito en otro CLR. Por ejemplo, puedes escribir una clase en C#, heredar de ella en VB.NET, y usarla en Python. Tienes la alternativa de escribir código en Mono en una variedad de lenguajes de programación.

- ✔ **Guiones y Embebidos**: La plataforma Mono también puede ser utilizada a través de guiones por tus aplicaciones al embeberse dentro de otras aplicaciones, para permitir que el código administrado y guiones se ejecuten en una aplicación nativa.

Contribuyendo al Proyecto

Existen diferentes formas de contribuir al Proyecto Mono, por ejemplo: reportar errores, contribuir con soluciones a éstos, escribir documentación o ejemplos, e inclusive escribir una aplicación basada en la Plataforma Mono, cualquier sea la opción es claro que siempre hay algo que hacer para este proyecto, la página[1] oficial lista una serie de tareas más concretas y la forma de lograrlas. La comunidad del Proyecto Mono siempre está deseosa de mejorar la Plataforma, por esta razón también desde el verano del año 2005 se lanza la convocatoria para que estudiantes de alrededor del mundo participen en el Google Summer of Code donde el Proyecto Mono sugiere mejoras y características que serán un proyecto para algún estudiante interesado. Además de contribuir en las listas de correos y foros en inglés, también existen grupos locales en español[10] para que la comunidad de habla hispana intercambie experiencia e ideas y provea soluciones a las dudas que alguien se pudiera plantear.

Conclusiones

La Plataforma Mono es una opción viable para aquellos desarrolladores que buscan tener una alta productividad, pues además de que se utiliza por defecto un lenguaje de alto nivel como C# y existe un recolector de basura automático para la administración de memoria, también hay una gran cantidad de documentación en el mercado, ejemplos y foros dedicados al desarrollo sobre esta plataforma. Además que tecnologías más recientes para crear aplicaciones con interfaces más amigables y más rápidas, como Silverlight 2, también están formando parte del proyecto.

Referencias

[1] http://www.mono-project.com/

[2] http://projects.gnome.org/tomboy/

[3] http://banshee-project.org/

[4] http://beagle-project.org/Main_Page



[5] http://f-spot.org/Main_Page

[6] http://do.davebsd.com/

[7] http://www.ecma-international.org/

[8] http://www.mono-project.com/Contributing

[9] http://www.mono-project.com/StudentProjects

[10] http://mono-project.com/User_Groups

Autor



Mario Carrion
Software Engineer
mario.carrion@gmail.com
http://www.mariocarrion.com/



Grails: Framework para el desarrollo de aplicaciones Web (4ta Parte)

Grails es un framework orientado al desarrollo de aplicaciones web de forma sencilla, rápida y divertida.



Una de la características principales que tiene cualquier aplicación es la posibilidad de poder obtener información variada en base a consultas realizadas a la base de datos, es por esta razón que en esta entrega veremos como realizar estas consultas.

Crear consultas

El ORM que posee **Grails** (GORM), soporta varias formas de poder realizar consultas, entre las que destacan:

- ✔ Dynamic Finders
- ✔ HQL
- ✔ Criteria
- ✔ Query-by-example

Donde cada una de estas formas tiene particularidades que las diferencian tanto por la facilidad de uso, como por la orientación que se da a cada una de ellas.

Dynamic Finders

Esta es una de las formas mas sencillas de hacer consultas, algunas de sus características son las siguientes:

- ✔ Traducen automáticamente las propiedades de la clase, para convertirse en parte de la consulta.
- ✔ Funcionan de similar forma a los filtros
- ✔ Se limitan a realizar consultas a una sola clase

Algunos ejemplos de este tipo de consultas, los mostramos a continuación:

Recupera la instancia de clase cuyo id es el 3

```
Expositor.get(3)
```

Cuenta el numero de objetos de la clase Evento

```
expositor.count()
```

Cuenta la cantidad de expositores cuya país de origen es Bolivia

```
Expositor.countByNacionalidad('Bolivia')
```

Listado completo de todos los objetos de la clase

```
Expositor.list()
```

Listado de Expositores

Id	Email	Materno	Nacionalidad	Nombre	Paterno
1	estebansaavedra@gmail.com	Lopez	Bolivia	Esteban	Saavedra
2	lucyayarde@gmail.com	Romero	Bolivia	Lucinda	Ayarde
3	clonwars@gmail.com	Guzman	Bolivia	Arnold	Guzman
4	frank.alcantara@gmail.com	Cohelo	Brasil	Frank	Alcantara
5	jperez@yahoo.com	Perez	Argentina	Juan	Perez
6	mporres@porres.com	Alvarez	Colombia	Marcos	Porres
7	kattheleen.saavedra@gmail.com	Ayarde	Brasil	kathleen	Saavedra
8	stephanie.saavedra@gmail.com	Ayarde	Colombia	Stephanie	Saavedra
9	scarlet.saavedra@gmail.com	Ayarde	Bolivia	Scarlet	Saavedra

Listado de todos los objetos de la clase siguiendo el orden especificado

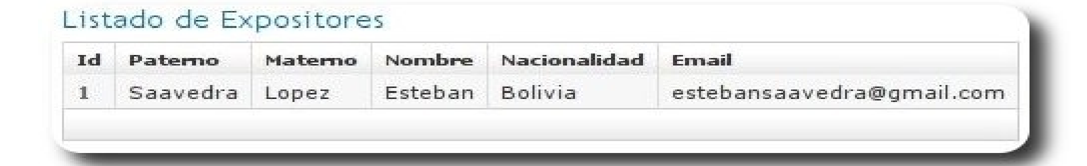
```
Expositor.listOrderByNacionalidad()
```

Listado de Expositores

Id	Paterno	Materno	Nombre	Nacionalidad	Email
5	Perez	Perez	Juan	Argentina	jperez@yahoo.com
1	Saavedra	Lopez	Esteban	Bolivia	estebansaavedra@gmail.com
2	Ayarde	Romero	Lucinda	Bolivia	lucyayarde@gmail.com
3	Guzman	Guzman	Arnold	Bolivia	clonwars@gmail.com
9	Saavedra	Ayarde	Scarlet	Bolivia	scarlet.saavedra@gmail.com
4	Alcantara	Cohelo	Frank	Brasil	frank.alcantara@gmail.com
7	Saavedra	Ayarde	kathleen	Brasil	kattheleen.saavedra@gmail.com
6	Porres	Alvarez	Marcos	Colombia	mporres@porres.com
8	Saavedra	Ayarde	Stephanie	Colombia	stephanie.saavedra@gmail.com

Devuelve el primer expositor cuyo pais de origen es Argentina y Bolivia

```
Expositor.findByNacionalidad ('Bolivia')
```



Listado de Expositores

Id	Paterno	Materno	Nombre	Nacionalidad	Email
1	Saavedra	Lopez	Esteban	Bolivia	estebansaavedra@gmail.com

Devuelve el primer expositor cuyo apellido paterno sea Ayarde y cuyo país de origen es Bolivia

```
Expositor.findWhere([ "paterno": "Ayarde", nacionalidad: "Bolivia"])
```



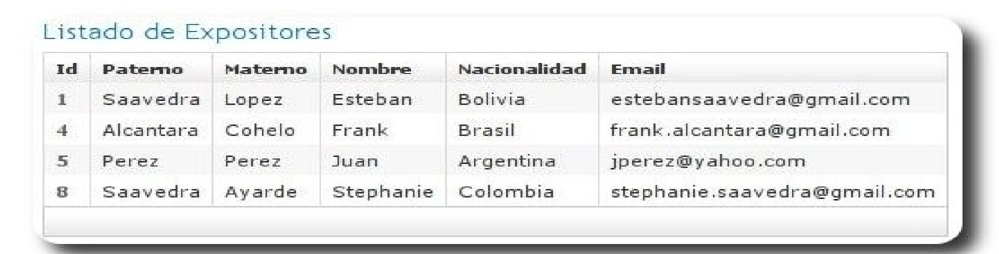
Listado de Expositores

Id	Paterno	Materno	Nombre	Nacionalidad	Email
2	Ayarde	Romero	Lucinda	Bolivia	lucyayarde@gmail.com

Otras opciones mas generales son: **findAllBy**, **findAllWhere**, **getAll** que tienen un funcionamiento análogo a los anteriores, con la diferencia que estos devuelven todas las instancias y no solo la primera instancia coincidente. Algunos ejemplos son mostrados a continuación:

Devuelve todos los eventos en cuyo nombre aparezca la palabra informática

```
Expositor.findAllByNombreLike("%an%")
```



Listado de Expositores

Id	Paterno	Materno	Nombre	Nacionalidad	Email
1	Saavedra	Lopez	Esteban	Bolivia	estebansaavedra@gmail.com
4	Alcantara	Cohelo	Frank	Brasil	frank.alcantara@gmail.com
5	Perez	Perez	Juan	Argentina	jperez@yahoo.com
8	Saavedra	Ayarde	Stephanie	Colombia	stephanie.saavedra@gmail.com

Devuelve todos los eventos en cuyo nombre aparezca la palabra Python y la fecha de inicio del evento tenga una antigüedad menor a un mes

```
Evento.findAllByNombreLikeAndIniciaDateGreaterThan("%Python%", new Date()-30)
```

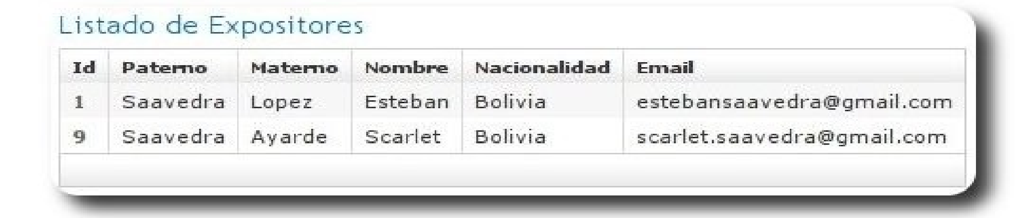
```
Expositor.findAllByNacionalidad ('Bolivia')
```



Listado de Expositores

Id	Paterno	Materno	Nombre	Nacionalidad	Email
1	Saavedra	Lopez	Esteban	Bolivia	estebansaavedra@gmail.com
2	Ayarde	Romero	Lucinda	Bolivia	lucyayarde@gmail.com
3	Guzman	Guzman	Arnold	Bolivia	clonwars@gmail.com
9	Saavedra	Ayarde	Scarlet	Bolivia	scarlet.saavedra@gmail.com

```
Expositor.findAllByNacionalidadAndPaterno ('Bolivia','Saavedra')
```



Listado de Expositores

Id	Paterno	Materno	Nombre	Nacionalidad	Email
1	Saavedra	Lopez	Esteban	Bolivia	estebansaavedra@gmail.com
9	Saavedra	Ayarde	Scarlet	Bolivia	scarlet.saavedra@gmail.com

el mismo efecto con la consulta

```
Expositor.findAllWhere([ "paterno": "Saavedra", nacionalidad: "Bolivia"])
```

En caso de desear implementar una paginación a los resultados obtenidos, podríamos hacer uso de la siguiente consulta:

```
Expositor.list(max: 10, offset: 20, sort: "nacionalidad", order "desc")
```

Consultas HQL

Si bien las consultas dinámicas son muy sencillas de utilizar y componer según los atributos de una clase, muchas veces en el desarrollo de aplicaciones precisamos contar con consultas mas complejas, para esto podemos hacer uso de HQL

Básicamente HQL es una forma ampliada y detallada de hacer consultas, y que puede ir acompañando a: **find**, **findAll** y **executeQuery**, como se muestra en el siguiente ejemplo

```
Expositor.findAll("from Expositor as e order by e.nacionalidad asc")
```

Listado de Expositores

Id	Paterno	Materno	Nombre	Nacionalidad	Email
5	Perez	Perez	Juan	Argentina	jperez@yahoo.com
1	Saavedra	Lopez	Esteban	Bolivia	estebansaavedra@gmail.com
2	Ayarde	Romero	Lucinda	Bolivia	lucyayarde@gmail.com
3	Guzman	Guzman	Arnold	Bolivia	clonwars@gmail.com
9	Saavedra	Ayarde	Scarlet	Bolivia	scarlet.saavedra@gmail.com
4	Alcantara	Cohelo	Frank	Brasil	frank.alcantara@gmail.com
7	Saavedra	Ayarde	kathleen	Brasil	kattheleen.saavedra@gmail.com
6	Porres	Alvarez	Marcos	Colombia	mporres@porres.com
8	Saavedra	Ayarde	Stephanie	Colombia	stephanie.saavedra@gmail.com

```
Expositor.findAll("from Expositor as e
                       where e.nombre = ?
                       and e.nacionalidad = ?
                       order by e.paterno asc", ["Kathleen", "Brasil"])
```

Listado de Expositores

Id	Paterno	Materno	Nombre	Nacionalidad	Email
7	Saavedra	Ayarde	kathleen	Brasil	kattheleen.saavedra@gmail.com

El mismo resultado puede ser obtenido por la siguiente consulta, donde la diferencia radica en el manejo de parámetros utilizados en la consulta.

```
Expositor.findAll("from Expositor as e
                       where e.nombre = :nombre
                       and e.nacionalidad = :nacionalidad
                       order by e.paterno asc", [nacionalidad :"Brasil", nombre :
"Kathleen"]
```

Como en los casos anteriores, **`findAll`** permite recuperar un conjunto de instancias y no solo la primera coincidente.

```
Expositor.findAll("From Expositor e", max: 10, offset: 20, sort: "nacionalidad", order
"desc")
```

executeQuery, presenta algunas diferencias a las anteriores consultas, por que no necesariamente debes recuperar todo el registro o la instancia de clase, tan solo puedes optar por recuperar el o los campos que sean necesarios, como en el ejemplo siguiente:

```
Expositor.executeQuery("select e.nombre from Expositor e where e.nacionalidad = ?",
"Bolivia")
```

Consultas en base a Criterios

Otro tipo de realizar consultas, es la creación de criterios, con la posibilidad de crear o incluir asociaciones. En los siguientes ejemplos mostramos como crear este tipo de consultas.

Devuelve los eventos que se iniciaron en los últimos 30 días y en cuyo nombre del evento figure la palabra Python

```
def hoy = new Date()
def eventos = Evento.withCriteria {
    like("nombre","%Python%")
    between("inicia",hoy-30, hoy)
    maxResults(10)}
```

Devuelve los eventos que se iniciaron en los últimos 30 días, además eventos asociados a conferencias realizadas en los últimos 7 días y en cuyo nombre figure la palabra **Grails**.

```
def hoy = new Date()
def resultado = Evento.withCriteria {
    conferencias {
        like("titulo","%Grails%")
        between("fecha", hoy-7, hoy)
    }
    between("inicia",hoy-30, hoy)
    maxResults(10)
```

Todo ser humano es libre si su libertad no atenta a la de los demás

Desarrollo

Devuelve los eventos donde cuyas conferencias se hayan realizado en los últimos 30 días.

```
def e = Evento.createCriteria()
def hoy = new Date()
def resultado = e.list{
    conferencias{
        between("fecha",hoy-30, hoy)
    }
}
```

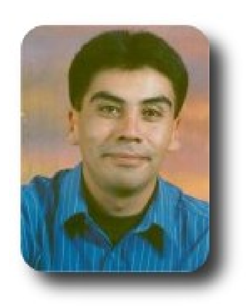
Referencias

[1] http://www.grails.org

[2] Getting Started with Grails

[3] The Definitive Guide to Grails

Autor



Esteban Saavedra López
Líder de la Comunidad ATIX (Oruro – Bolivia)
Activista de Software Libre en Bolivia
jesaavedra@opentelematics.org
http://jesaavedra.opentelematics.org

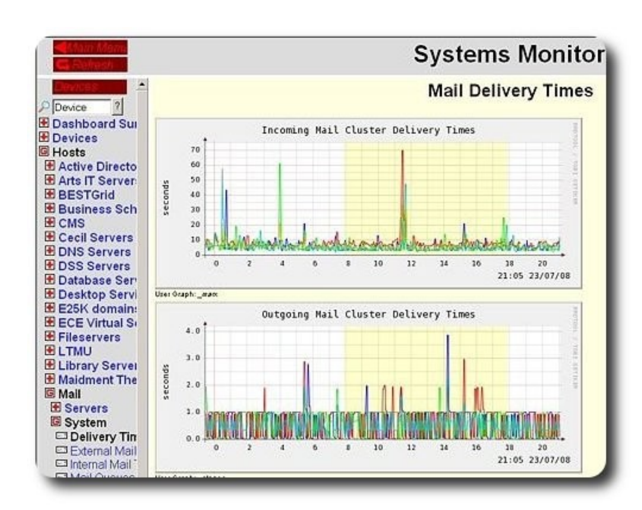


Mejorando MRTG con Router2

En el anterior número de esta revista vimos la forma de como monitorear servidores mediante la herramienta MRTG, ahora presentaremos una mejora al manejo de estas gráficas mediante un Front-End llamado ROUTERS2.

Introducción

La herramienta MRTG nos entrega gráficas muy útiles de los sistemas que estamos monitoreando, pero para poder ver estas gráficas debemos ingresar a distintas páginas web (una página web por cada parámetro monitoreado de cada sistema) lo cual se nos complica más al momento de monitorear varios parámetros de varios sistemas, debido a que tendremos muchas páginas web como resultado y con diferentes URLs.



ROUTERS2 nos da la posibilidad de agrupar el acceso a todas estas gráficas desde un solo front-end, es decir desde una sola página web. Mejorando así la administración de este monitoreo.

Prerequisitos

- libart_lgpl: Librerías Libart.
- libart_lgpl-devel: Librerías de desarrollo Libart.
- zlib: Librería de compresión de data.
- zlib-devel: Librería de desarrollo de compresión de data.
- libpng: Librerías para PNG (Portable Network Graphics).
- libpng-devel: Librerías de desarrollo para PNG (Portable Network Graphics).
- freetype: Biblioteca que implementa font-engine.
- freetype-devel: Biblioteca de desarrollo que implementa font-engine.
- perl: Lenguaje de programación PERL.
- apache: Web server
- rrdtool: Base de Datos rrdtool (Round Robin Database Tool).
- rrdtool-devel: Librería de desarrollo para rrdtool.
- perl-rrdtool: Librería perl para rrdtool.

Consideraciones en la configuración de Apache

Para este ejemplo el archivo de publicación de Apache es el **`/var/www/html`** y el directorio de los **`cgi-bin`** es el **`/var/www/cgi-bin`**.

Consideraciones en la configuración de MRTG

Para este ejemplo el directorio con los archivos de configuración del MRTG es el **`/etc/mrtg`** y el directorio donde se generan los archivos de salida del MRTG es el **`/mrtg`**.

En cada archivo de configuración de cada servidor se debe especificar la utilización del RRDTOOL y de las librerías Perl (para mayor información sobre la configuración del MRTG pueden consultar el artículo **Monitoreo de servidores mediante MRTG** publicado en el número anterior de esta revista), a continuación mostramos un ejemplo de archivo de configuración del MRTG llamado **`myPC.cfg`**:

```
EnableIPv6: no
WorkDir: /mrtg
LogFormat: rrdtool
PathAdd: /usr/bin
LibAdd: /usr/lib/perl5/vendor_perl/5.8.5/i386-linux-thread-multi
Target[myPC_ram]: 1.3.6.1.4.1.2021.4.6.0&1.3.6.1.4.1.2021.4.5.0:public@172.16.16.1:
AbsMax[myPC_ram]: 3076140
MaxBytes[myPC_ram]: 3076140
Title[myPC_ram]: RAM Libre de myPC
Legend1[myPC_ram]: RAM Libre
Legend2[myPC_ram]: RAM TOTAL
LegendI[myPC_ram]: RAM Libre: 
LegendO[myPC_ram]: RAM TOTAL: 
Ylegend[myPC_ram]: Memoria RAM
Options[myPC_ram]: gauge,absolute,integer,nopercent
PageTop[myPC_ram]: <H1>RAM Libre de myPC</H1>
ShortLegend[myPC_ram]: Bytes
kilo[myPC_ram]: 1024
kMG[myPC_ram]: k,M,G,T,P
Target[myPC_cpu]:
1.3.6.1.4.1.2021.10.1.3.1&1.3.6.1.4.1.2021.10.1.3.1:public@172.16.16.1:
AbsMax[myPC_cpu]: 100
MaxBytes[myPC_cpu]: 100
Title[myPC_cpu]: CPU Utilizado en myPC
Legend1[myPC_cpu]: CPU Utilizado
Legend2[myPC_cpu]: CPU Utilizado
LegendI[myPC_cpu]: CPU Utilizado: 
LegendO[myPC_cpu]: CPU Utilizado: 
Ylegend[myPC_cpu]: CPU Utilizado %
ShortLegend[myPC_cpu]: %
Options[myPC_cpu]: gauge,absolute,integer,nopercent
PageTop[myPC_cpu]: <H1>CPU Utilizado en myPC</H1>
Target[myPC_disk]:
1.3.6.1.4.1.2021.9.1.9.1&1.3.6.1.4.1.2021.9.1.9.1:public@172.16.16.1:
AbsMax[myPC_disk]: 100
MaxBytes[myPC_disk]: 100
Title[myPC_disk]: DISCO Utilizado en myPC
Legend1[myPC_disk]: DISCO Utilizado
Legend2[myPC_disk]: DISCO Utilizado
LegendI[myPC_disk]: DISCO Utilizado: 
LegendO[myPC_disk]: DISCO Utilizado: 
Ylegend[myPC_disk]: DISCO Util. %
ShortLegend[myPC_disk]: %
Options[myPC_disk]: gauge,absolute,integer,nopercent
PageTop[myPC_disk]: <H1>DISCO Utilizado en myPC</H1>
```

Instalación

Se debe obtener la última versión del ROUTERS2 de **http://www.steveshipway.org/software/**

El archivo de instalación que conseguiremos estará comprimido, al descomprimir este archivo obtendremos el directorio **~/routers2-vX.XX**, dentro de éste encontraremos el archivo **install.pl;** para la instalación debemos ejecutar el siguiente comando:

```
# perl install.pl
```

Donde se tienen las siguientes salidas como preguntas:

```
0. Attempting to identify your OS and web server...
   - I think you are running under UNIX.
   - I think you have Apache installed in
     /etc/httpd/conf
1. Web server document root directory.
   This is the full path of the base document directory of your web server.
   Document root [/var/www/html]? #
2. Web server CGI directory.
   This is the full path of the directory where your web server keeps the
   CGI scripts.
   CGI directory [/var/www/cgi-bin/]? #
3. MRTG config file directory.
   This is the full path of the directory where your MRTG configuration files
   are kept
   MRTG config directory? #/etc/mrtg
4. MRTG config files.
   This is the wildcarded filename format for your MRTG configuration files.
   Use a '*' to mean 'any characters' - for example, '*.cfg' or '*/*.conf'.
   MRTG files [*.cfg]? #
5. RRD Database directory.
   This is the full path of the directory where your .rrd files are kept
   RRD directory [/tmp]? # /mrtg
6. Perl executable.
   This is the full pathname of the Perl executable file.
   Perl executable [/usr/bin/perl]? #
7. routers2.cgi configuration file
   This is the file that will hold the routers2.cgi configuration.  Unless you
   have a reason to move it, stick with the default.
   If this file already exists, I will ask before overwriting it!
   Configuration file [/mrtg/routers2.conf]? #

ASKING OPTIONS

1. Net::SNMP does not appear to be installed.  Routing table extensions
   have been disabled.
   If you subsequently install Net::SNMP, then you can enable the extensions
   in the routers2.conf file.
   GD Perl Library is detected.
2. The Compact Summary pages will be enabled.
3. How big should 1K and 1M be?  This is the 'usebigk' parameter from the
   routers2.conf file.  You have three options - 'yes', 'no' and 'mixed'.
   yes   -> 1K=1024, 1M=1024x1024
   no    -> 1K=1000, 1M=1000x1000
   mixed -> 1K=1024, 1M=1024x1000
   'usebigk' option [mixed]? #
```

```
4. Do you want to use authentication?  You can always enable this later if
   you change your mind.  There are other options available in the
   configuration file as well, so you should check.  If you are unsure, select
   the default.
   none -> do not use any additional authentication (default)
   http -> use web server's own authentication, if available
   ldap -> use ldap/ldaps authentication
   file -> use a password file (not recommended)
   auth option [none]? #
5. Caching option
   routers2 has support for fast CGI utilities such as speedycgi and mod_perl.
   It achieves this by data caching between invocations.
   This can dramatically improve performance on systems with a large
   number of .cfg files, however it slows performance if you do not have
   these features.  If you are unsure, answer NO.
   Valid answers: no, modperl, speedycgi
   Caching option [no]? #
6. Can I attempt to send an email to the author to let him know that the
   software has been installed?  This will only give your routers.cgi version,
   Perl version, and Operating System version.
   Can I mail [no]? #

INSTALLING SOFTWARE

Perl is       : /usr/bin/perl
MRTG files    : /etc/mrtg/*.cfg
RRD files     : /mrtg
Doc root      : /var/www/html
CGI bin       : /var/www/cgi-bin/
Config file : /mrtg/routers2.conf
Routingtable: INACTIVE
Compact page: ENABLED
Caching       : DISABLED
'usebigk'     : mixed
Auth option : NONE
Mail Steve    : no
Other options can be set later by modifying the Config file
Continue to install [no]? # yes
** ALL COMPLETE **
```

Esta configuración inicial al momento de la instalación, es guardada en el archivo de configuración del ROUTERS2, este archivo es el **`/mrtg/routers2.conf`** el cual puede ser modificado en caso de ser necesario.

Una vez instalada la herramienta ROUTERS2, ésta tomará automáticamente como referencia a los servidores que se estén monitoreando por el MRTG y que tengan sus archivos .cfg en el directorio **`/etc/mrtg`**, y los desplegará en la página web front-edm. En este ejemplo solo se tiene el archivo **`myPC.cfg`** por tal motivo solo este equipo será mostrado en el front-end, pero en el caso de agregar en el monitoreo del MRTG más servidores, equipos de comunicación, etc., la herramienta ROUTERS2 agregará automáticamente a estos equipos en la página web front-end.

Front-end de monitoreo

Para ingresar a la página web del front-End, con un web browser debemos ingresar al siguiente URL: **`http://servidor/cgi-bin/routers2.cgi`** donde el servidor es el mismo donde se instaló el ROUTERS2.

A continuación mostramos la página web inicial del front-edn, en la cual se puede observar en la columna izquierda al servidor myPC como equipo monitoreado; como se explicó antes solo tenemos configurado en el MRTG a este equipo, si se tuviesen más equipos monitoreados

aparecerían debajo de myPC:

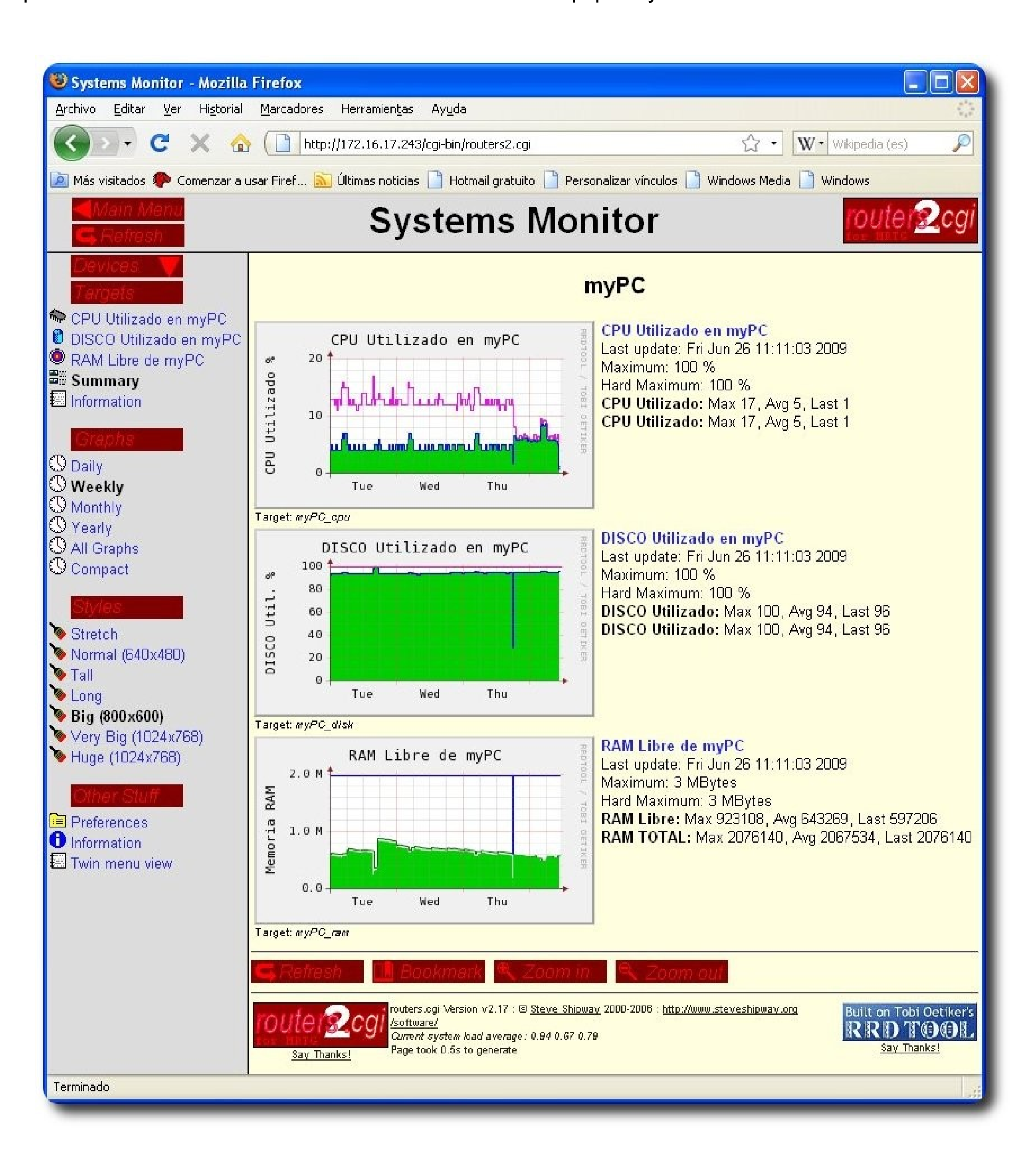


Si le damos un clic a myPC podremos observar el primer parámetro monitoreado en la parte derecha:

Si le damos un clic a OPTIONS podremos ver la parte más interesante de esta herramienta y podremos escoger:

- En **Targest** están todos los parámetros monitoreados de este servidor: CPU, DISCO y RAM, y el sumario que contienen a todos estos parámetros.
- En **Graphs** están los períodos de tiempo: Diario, Semanal, mensual, Anual.
- En **Styles** podremos ver varios tipos de tamaños de los gráficos.

A continuación veamos un ejemplo en donde se puede apreciar el sumario de todos los parámetros monitoreados en la última semana del equipo myPC:



Conclusiones

Esta herramienta mejora la administración del monitoreo del MRTG desde una sola página web.

Referencias

[1] http://www.steveshipway.org/software/

Autor



Renzo Martinez Pardo
Ingeniero Sistemas Electrónicos
renzomp@lycos.com



Levantar un servidor Nagios en Ubuntu

En este artículo, se mostrará como poner en funcionamiento un Servidor Nagios en Ubuntu, siendo ésto de gran ayuda para los administradores de Sistemas, porque de esta manera podrán tener conocimiento de todo el Hardware puesto en red en tiempo real.

Introducción

Nagios es un sistema libre para la monitorización de redes (y Servidores) ampliamente utilizado, que vigila los equipos (hardware) y servicios (software) que se especifiquen, alertando cuando el comportamiento de los mismos no sea el deseado.

Características

Entre sus características principales figuran:

- ✔ Monitorización de servicios de red como ser SMTP, POP3, HTTP, SNMP, FTP, etc
- ✔ Monitorización de los recursos de sistemas hardware como ser carga del procesador, uso de los discos, memoria, estado de los puertos, etc.)
- ✔ Independencia de sistemas operativos,
- ✔ Monitorización remota mediante túneles SSL cifrados ó SSH
- ✔ Posibilidad de programar plugins específicos para nuevos sistemas.

Nagios proporciona una gran versatilidad para consultar prácticamente cualquier parámetro de interés de un sistema, y genera alertas, que pueden ser recibidas por los administradores mediante despliegue de información en pantalla, correo electrónico y/o mensajes SMS, cuando estos parámetros exceden de los márgenes definidos por el administrador.

Nagios fue originalmente diseñado para ser ejecutado en GNU/Linux, pero también se ejecuta bien en variantes de Unix. Está licenciado bajo la GNU General Public License Version 2.

Instalación y Configuración del Servidor Nagios

Primero instalaremos Apache2 y las librerías necesarias para utilizar Nagios.

Instalamos Apache2 y el paquete essential con librerías para compilar y desarrollar:

```
#apt-get install apache2
#apt-get install build-essential
```

Instalamos la librería gd2:

```
#apt-get install libgd2-xpm-dev
```

Creamos una cuenta de usuario 'nagios' y modificamos la contraseña:

```
#useradd -m nagios
#passwd nagios
```

Crearemos la cuenta de usuario para comandos externos suministrados a través de la interfaz web. Añadiremos al usuario '`nagcmd`' al grupo '`nagios`' y al '`www-data`':

```
#groupadd nagcmd
#usermod -a -G nagcmd nagios
#usermod -a -G nagcmd www-data
```

Ahora vamos al directorio **/home/mi_home/** para realizar las descargas de nagios y los plugins de la página **http://www.nagios.org/download/**:

```
#cd /home/mi_home/
#wget http://osdn.dl.sourceforge.net/sourceforge/nagios/nagios-3.0.6.tar.gz
#wget http://osdn.dl.sourceforge.net/sourceforge/nagiosplug/nagios-
plugins-1.4.11.tar.gz
```

Desempaquetamos el tar y accedemos a la carpeta del mismo:

```
#tar xzf nagios-3.0.6.tar.gz
#cd nagios-3.0.6
```

Ejecutamos el script para configurar con los siguientes parámetros:

```
#./configure --with-command-group=nagcmd
```

Compilamos el código fuente de nagios e instalamos:

```
#make all
#make install
#make install-init
#make install-config
#make install-commandmode
```

Personalizamos nuestra configuración insertando nuestra dirección de correo para que nagios se pueda comunicar con nosotros y podamos recibir las alertas en el fichero **/usr/local/nagios/etc/objects/contacts.cfg**

```
#vi
/usr/local/nagios/etc/objects/contacts.cf
g
```

Configuramos la interfaz Web e instalamos el fichero de configuración de Apache2 en **conf.d**:

```
#make install-webconf
```

Creamos la cuenta de usuario '**nagiosadmin**' para hacer login al interfaz web:

```
#htpasswd -c
/usr/local/nagios/etc/htpasswd.users
nagiosadmin
```

Recargamos Apache2 para que modifique los cambios realizados:

```
#/etc/init.d/apache2 reload
```

Desempaquetamos los plugin e ingresamos en el directorio:

```
#cd /home/mi_home/
#tar xzf nagios-plugins-1.4.11.tar.gz
#cd nagios-plugins-1.4.11
```

Antes de compilar e instalar los plugins, nagios por defecto no tiene los plugins de SNMP (los cuales son muy importantes para revisar equipos por la red), entonces debemos bajarlos de su web: **http://www.net-snmp.org/download.html** (ojo, son paquetes binarios en rpm).

Los copiamos al directorio donde desempaquetamos los plugins.

Compilamos e instalamos los plugins:

```
#./configure --with-nagios-user=nagios --
with-nagios-group=nagios
#make
#make install
```

Configuramos Nagios para que arranque automáticamente:

```
#ln -s /etc/init.d/nagios
/etc/rcS.d/S99nagios
```

Verificamos que la configuración que tenemos es correcta con el siguiente comando:

```
#/usr/local/nagios/bin/nagios -v
/usr/local/nagios/etc/nagios.cfg
```

Finalmente, y si todo ha salido bien, procedemos a arrancar Nagios:

```
#/etc/init.d/nagios start
```

Con todo ésto, ya tenemos instalado y configurado Nagios para funcionar y recibir alarmas del mismo servidor nagios (localhost). Abrimos un explorador e ingresamos la dirección '**`http://localhost/nagios/`**' (o **`http://IP_maquina/nagios/`**) y cuando nos pida autenticación, ingresamos el usuario creado anteriormente '**`nagiosadmin`**' con su correspondiente contraseña.

Una vez dentro, podemos realizar un click en "Service Detail" y monitorizar nuestro servidor. Recordatorio: La ubicación de todos los ficheros de configuración están ubicados en **`/usr/local/nagios`**.

Entonces, hasta ahora tenemos un servidor nagios, levantado, pero solo podemos ver los servicios de nuestro localhost (servidor o desktop) con el nagios instalado; debemos hacer ciertas configuraciones para poder monitorear servicios de hardware de Red, así como de otros Servidores (en Linux, Unix o Windows) y podemos monitorizar varias cosas, como ser uso de la CPU, uso de Memoria, uso de Disco Duro, etc.

Monitorizando un Servidor Linux



Para monitorizar otros servidores o máquinas con Linux que no sea nuestra Localhost (Servidor o máquina en la cual tenemos instalado Nagios), debemos usar el plugin NRPE, este plugin, tiene dos partes:

El plugin **`NRPE check_nrpe`** que reside en el Servidor Nagios. El demonio **`NRPE`**, que reside en la máquina a monitorizar.

La forma en que funcionan es la siguiente:

- ✔ Nagios ejecuta **`check_nrpe`** y le dice que servicio necesita revisar.
- ✔ **`check_nrpe`** se contacta con el demonio NRPE en el host remoto (una opción es hacerlo por SSL).
- ✔ El demonio **`NRPE`** corre y revisa el servicio requerido
- ✔ El resultado es pasado del demonio NRPE a **`check_nrpe`** y éste enviado y procesado a Nagios.

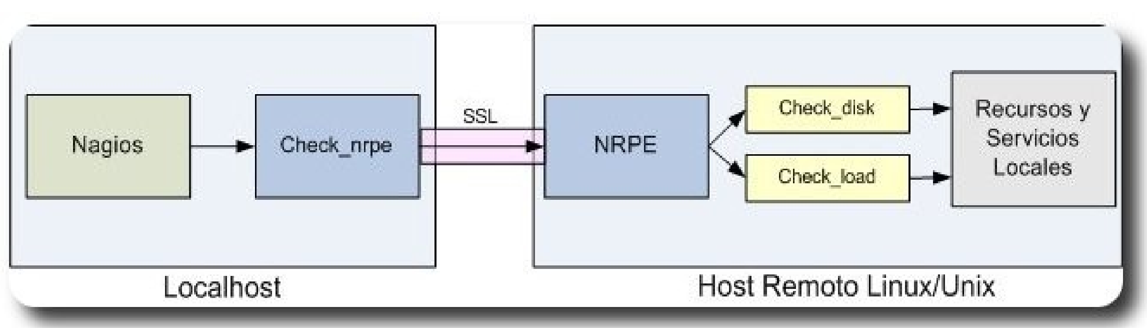


Diagrama 1: Funcionamiento de NRPE

Fuente: Nagios 3.x Documentation

Ahora continuaremos con la Instalación del Host Remoto.

Entramos a la consola y tomamos privilegios de root.

```
$sudo -s
```

creamos un usuario nagios con su contraseña:

```
#/usr/sbin/useradd nagios
#passwd nagios
```

Instalamos los plugins de nagios (los mismos que instalamos en el Servidor Nagios).

```
#cd /home/mi_home/
#wget
http://osdn.dl.sourceforge.net/sourceforg
e/nagiosplug/nagios-plugins-1.4.11.tar.gz
#tar xzf nagios-plugins-1.4.11.tar.gz
#cd nagios-plugins-1.4.11
```

Compilamos e Instalamos los plugins

```
#./configure
#make
#make install
```

Los permisos al directorio del plugin deben ser configurados de la siguiente manera:

```
#chown nagios.nagios /usr/local/nagios
#chown -R nagios.nagios
```

Ahora instalamos en demonio NRPE:

```
#cd ~/downloads
#wget
http://osdn.dl.sourceforge.net/sourceforg
e/nagios/nrpe-2.8.tar.gz
#tar xzf nrpe-2.8.tar.gz
#cd nrpe-2.8
```

Lo compilamos

```
#./configure
#make all
```

Instalamos el pulgin **NRPE** (para pruebas), el demonio y el archivo de configuración del demonio

```
#make install-plugin
#make install-deamon
#make install-deamon-config
```

Instalamos el demonio **NRPE** como un servicio en **xinetd**

```
#make install-xinetd
```

Editamos el archivo **/etc/xinetd.d/nrpe** y le añadimos la direccion IP del servidor Nagios en la línea only_from

```
   only_from=192.168.0.205
<nagios_ip_address>
```

Añadimos la siguiente entrada del demonio **NRPE** a **/etc/services**

```
nrpe          5666/tcp         #nrpe
```

Reiniciamos el servicio ninetd

```
#/etc/init.d/xinetd restart
```

Ahora probamos de manera local el demonio **NRPE**

primero verificamos que el demonio **NRPE** esté corriendo en **xinetd**

```
#netstat -at | grep nrpe
```

la salida debe ser:

```
tcp   0    0 *:nrpe          *:*      LISTEN
```

Si esa es la salida, perfecto, está funcionando el demonio. Si no, se debe revisar lo siguiente: haber añadido nrpe en el

archivo **/etc/services** Que la directiva **only_from** rn **/etc/xinetd.d/nrpe** contenga una direccion IP válida, que xinetd esté instalado y corriendo

Ahora, revisamos que el demonio **NRPE** esté funcionando correctamente

```
#/usr/local/nagios/libexec/check_nrpe -H
localhost
```

Deberíamos tener el siguiente resultado

```
NRPE v2.8
```

Ahora nos aseguramos de que el firewall deje que el demonio **NRPE** sea accesado de forma remota abriendo el puerto tcp 5666 dentro de iptables

Cabe recalcar que se puede editar el archivo de configuración por el cual corre el nrpe en la máquina local:

```
#vi /usr/local/nagios/etc/nrpe.cfg
```

Así como está la actualización lo que hará el NRPE es el monitoreo de usuarios, la carga del procesador, uso del disco duro, listado de todos los procesos y ver si existen procesos **Zombies**

Ahora, pasemos a la configuración del Servidor Nagios:

debemos instalar el plugin check_nrpe

```
$sudo -s
```

Vamos al directorio con los archivos bajados

```
#cd  ~/downloads
#wget
http://osdn.dl.sourceforge.net/sourceforg
e/nagios/nrpe-2.8.tar.gz
```

Extraemos el NRPE

```
#tar xzf nrpe-2.8.tar.gz
#cd nrpe-2.8
```

Compilamos el NRPE

```
#./configure
#make all
```

instalamos el plugin

```
#make install-plugin
```

Nos aseguramos que el **check_nrpe** se puede comunicar con el demonio NRPE en el linux remoto, (La IP de mi máquina linux remota es 192.168.0.2)

```
#/usr/local/nagios/libexec/check_nrpe -H
192.168.0.2
```

Debería respondernos:

```
NRPE v2.8
```

Si no da esa respuesta, debemos revisar que el firewall del host remoto no nos bloquee, verificar que el demonio esté instalado en **xinetd** (Ver todo eso más arriba)

Ahora debemos crear definiciones de comandos (command definitions), para eso editamos el commands.cfg Nagios ejecuta **check_nrpe** y le dice que servicio necesita revisar.

check_nrpe se contacta con el demonio NRPE en el host remoto (una opción es hacerlo por SSL.

El demonio **NRPE** corre y revisa el servicio requerido

El resultado es pasado del demonio **NRPE** a **check_nrpe** y éste enviado y procesado a Nagios.

```
#vi /user/local/nagios/etc/commands.cfg
```

y añadimos:

```
   define command{
        command_name    check_nrpe
        command_line    $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
        }
```

Ya estamos listos para añadir servicios que serán monitorizados por la configuración de Nagios, entonces creamos las definiciones del servicio para máquinas Linux/Unix, ésto creando un archivo con todo lo que se pondrá a continuación en:

```
   #vi /usr/local/nagios/etc/objects/linux-box.cfg
```

Donde ponemos:

```
   Define host{
        name                            linux-box ;name of this template
        use                             generic-host; Inherit default values
        check_period                    24x7
        check_interval                   5
        retry_interval                  1
        max_check_attemps               10
        check_command                    check-host-alive
        notification_period             24x7
        notification_interval           30
        notification_options            d, r
        cotact_groups                   admins
        register                        0; DONT REGISTER THIS – ITS A TEM
        }
```

Cabe recalcar que las definiciones que usa este template (linux-box), hereda los valores por defecto del **template generic-hosts**, el cual está definido en **localhost.cfg**

Ahora definimos el el nuevo host (podemos definir varios hosts, dependiendo el número de servers Linux que necesitamos monitorear)

```
   define host{
        use         linux-box ;Inherit default values from tempate
        host_name   host_remoto_1 ;The name we're giving this server
        alias       Firewall ;A longer name for the Server
        address     192.168.0.1 ;IP address of the Server
        }
```

Ahora definimos las funciones que se monitorizarán, por ejemplo empecemos con el CPU. Recuerden que check_load es enviado a **check_nrpe** y éste le dice al demonio **NRPE** que corra un comando **check_load** definido en el archivo **nrpe.cfg;** ésto se hace a todos los servicios que revisemos que estén dentro del **nrpe.cfg**

```
define service {
     use                        generic-service
     host_name                  host_remoto_1
     service_description        CPU Load
     check_command              check_nrpe!check_load
     }
```

Para saber el número de usuarios conectados:

```
define service{
     use                        generic-service
     host_name                  host_remoto_1
     service_description        Current Users
     check_command              check_nrpe!check_users
     }
```

Para saber el espacio de disco en /dev/hda1

```
define service{
     use                        generic-service
     host_name                  host_remoto_1
     service_description        /dev/hda1 Free Space
     check_command              check_nrpe!check_hda1
     }
```

Para saber el número total de procesos

```
define service{
     use                        generic-service
     host_name                  host_remoto_1
     service_description        Total Processes
     check_command              check_nrpe!check_total_procs
     }
```

Para saber el número de procesos zombies

```
define service{
     use                        generic-service
     host_name                  host_remoto_1
     service_description        Zombie Processes
     check_command              check_nrpe!check_zombie_procs
     }
```

Finalmente, añadimos la siguiente línea al archivo **`nagios.cfg`**

```
cfg_file=/usr/local/nagios/etc/objects/linux-box.cfg
```

Esto para que nagios sepa que cosas hacer correr.

Revisamos que todo esté bien

```
#/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Si hay errores, los arreglamos, si todo está bien, reiniciamos Nagios

```
#/etc/init.d/nagios restart
```



Listo, esa es la configuración para monitorizar Servidores Linux de forma básica, también se pueden añadir más comandos como el **check_swap**

Debemos ingresar al host remoto como usuario nagios (que se creo en la instalación del NRPE), y digamos que queremos que nos de un Warning con el 25% de espacio disponible y que nos de un critical con el 10% entonces:

```
#/usr/local/nagios/libexec/check_swap -w 25% -c 10%
```

entonces, añadimos ese comando al nrpe:

```
#vi  /usr/local/nagios/etc/nrpe.cfg
```

y añadimos:

```
command[check_swap]=/usr/local/nagios/libexec/check_swap -w 25% -c 10%
```

Ahora, volvemos a nuestro servidor Nagios y ahí definimos el servicio en **linux-box.cfg**

```
   define service{
        use                     generic-service
        host_name               host_remoto_1
        service_description     Swap Usage
        check_command           check_nrpe!check_swap
        }
```

Listo, nos aseguramos que no hayan errores y reiniciamos nagios

```
#/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
#/etc/init.d/nagios restart
```

Monitorizando Servidores en Windows



Ahora, como se vió en algún momento, no solo podemos monitorizar máquinas en Linux/Unix, sino también en Windows.

Nagios recomienda usar el plugin **NSClient++**, así como en Linux hay que instalar ciertas cosas en la máquina a ser monitorizada, lo mismo pasa con Windows.

La forma de funcionar es la siguiente:

- En los paquetes de plugin de Nagios viene por defecto el check_nt,
- check_nt se comunica con el host remoto (más específicamente al NSClient++)
- NSClient++ es el que revisa información como ser CPU, Memoria, Espacio de Disco, etc.

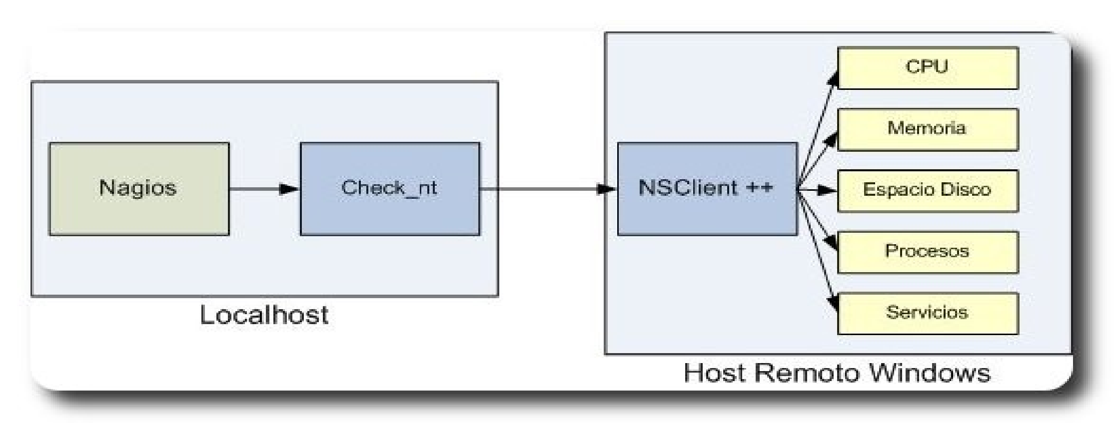


Diagrama 2: Funcionamiento de Check_nt y NSClient ++

Fuente: Nagios 3.x Documentation

Ahora pasamos a la configuración, para lo cual primero debemos abrir para editar:

```
#vi /usr/local/nagios/etc/nagios.cfg
```

Ahí descomentamos (borramos) numeral (#) de la siguiente línea.

```
#cfg_file=/usr/local/nagios/etc/objects/windows.cfg
```

Ahora debemos instalar el agente en Windows, para eso vamos y descargamos de la página **http://sourceforge.net/projects/nscplus**.

Descargamos la última versión, la que es RC (release candidate), ya que hice pruebas con la mayoría, y el último release candidate fue el que menos bugs tiene (**NSClient 0.3.6 RC8**)

Lo Instalamos (una interfaz gráfica), elegimos que la carpeta sea **C:\NSClient++**, todas las demás opciones las dejamos en blanco.

Una vez instalado, abrimos el archivo **NSC.INI** y lo editamos:

Descomentando (borrando) todos los módulos en la sección [modules], menos **CheckWMI.dll**

En la sección **[setting]**, ponemos un password, ya que si se lo deja en blanco, el bug se activa y el agente no funciona.

En la misma sección **[setting]**, si queremos dejamos en blanco la ip de **allowed_hosts** (para que cualquier server se conecte) o ponemos la IP del Servidor Nagios (mejor).

Y en la sección **[NSClient]**, nos aseguramos de descomentar el puerto y que esté en 12489

Guardamos los cambios y Salimos

Ahora paramos el servicio con

```
nsclient++ /stop
```

y lo iniciamos con

```
nsclient++ /start
```

Ahora todos los cambios surtieron efecto.

Seguidamente definimos los objetos editando el **`windows.cfg`** en

```
#vi
/usr/local/nagios/etc/objects/windows.cfg
```

Donde ponemos:

```
   define host{
      use       windows-server  ; Inherit default values from a Windows server template
      host_name winserver ;Nombre que le damos al host
      alias     Servidor Windows SQL
      address   192.168.0.2
      }
```

Listo, ahora podemos añadir definiciones de servicio en el mismo archivo

Monitoreamos la versión del NSClient

```
   define service{
        use                     generic-service
        host_name               winserver
        service_description     NSClient++ Version
        check_command           check_nt!CLIENTVERSION
        }
```

Monitoreamos el tiempo que el Servidor está “en línea” (up time)

```
   define service{
        use                     generic-service
        host_name               winserver
        service_description     Uptime
        check_command           check_nt!UPTIME
        }
```

Monitorizamos la carga del CPU, donde nos dará un Warning al 80% y Crítico al 90% si se mantienen con esa carga por más de 5 minutos

```
   define service{
         use                    generic-service
         host_name              winserver
         service_description    CPU Load
         check_command          check_nt!CPULOAD!-l 5,80,90
         }
```

Monitorizamos el uso de memoria, donde nos da un warning con el 80% alcanzado un Critical con el 90 %

```
   define service{
         use                    generic-service
         host_name              winserver
         service_description    Memory Usage
         check_command          check_nt!MEMUSE!-w 80 -c 90
         }
```

Monitorizamos el uso del Disco **C:** (nota, ésto se puede hacer para todos los discos que necesitemos)

```
define service{
      use                     generic-service
      host_name               winserver
      service_description     C:\ Drive Space
      check_command           check_nt!USEDDISKSPACE!-l c -w 80 -c 90
     }
```

Finalmente revisamos si el proceso Explorer.exe está corriendo

```
define service{
     use                     generic-service
     host_name               winserver
     service_description     Explorer
     check_command           check_nt!PROCSTATE!-d SHOWALL -l Explorer.exe
     }
```

Ahora, le ponemos el password al archivo **commands.cfg** para que el **NSClient** pueda ser accedido:

```
#vi /usr/local/nagios/etc/objects/commands.cfg
```

Donde:

```
define command{
    command_name check_nt
    command_line $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -s \
    Contraseña_Configurada -v $ARG1$ $ARG2$
    }
```

Listo, nos aseguramos que no hayan errores y reiniciamos nagios

```
#/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
#/etc/init.d/nagios restart
```

Monitorizando Hardware de Red (Hubs, Switches y/o Routers)

Ahora pasaremos al monitoreo de Hardware de Red, como ser Switches y Routers. Cabe recalcar que este hardware debe poder soportar SNMP, generalmente, el hardware al que se le asigna una direccion IP, soporta SNMP, esto es importante ya que se usa este protocolo y plugins del net-SNMP (que se instalaron al principio del tutorial).

En caso de que nuestro Hardware no Soporte SNMP, verificar si con un upgrade al firmware lo puede hacer y si no, solo nos limitaremos a saber que está en línea mediante PING.

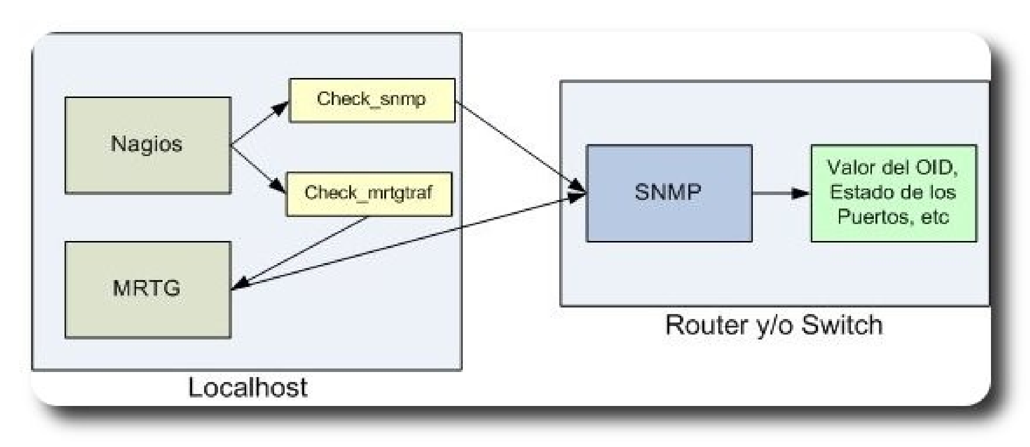


Diagrama 3: Funcionamiento de Check_snmp y Check_mrtgtref

Fuente: Nagios 3.x Documentation

Primero debemos editar:

```
#vi /usr/local/nagios/etc/nagios.cfg
```

ahí, descomentamos borrando el numeral (#) de la siguiente fila

```
#cfg_file=/usr/local/nagios/etc/objects/switch.cfg
```

Ahora, debemos crear la definición del objeto para monitorear un switch o un router abrimos para editar

```
#vi /usr/local/nagios/etc/objects/switch.cfg
```

y ahí modificamos

```
   define host{
           use        generic-switch ; Inherit default values from a template
           host_name  Switch_1 ; The name we're giving to this switch
           alias      TrendNet TEG-240WS ; A longer name associated with the switch
           address    192.168.0.20  ; IP address of the switch
           hostgroups allhosts,switches  ; Host groups this switch is associated with
          }
```

(hay que añadir la cantidad de switches, routers o hubs que tengamos en esta sección, un “define host” para cada uno)

Listo, ahora podemos añadir servicios a ser monitorizados, como ser la perdida de paquetes y el Round Trip Average:

```
define service{
    use       generic-service  ; Inherit values from a template
    host_name Switch_1 ; The name of the host the service is associated with
    service_description   PING ; The service description
    check_command  check_ping!200.0,20%!600.0,60% ; Command used to monitor the Srv
    normal_check_interval 5 ; Check the service every 5 minutes under nor. Cond.
    retry_check_interval 1 ; Re-check the service every minute until its final/hard
    state is determined
            }
```

Si el hardware soporta SNMP, podemos monitorear el estatus de dicho protocolo:

```
define service{
     use                        generic-service ; Inherit values from a template
     host_name                  Switch_1
     service_description        Uptime
     check_command              check_snmp!-C public -o sysUpTime.0
    }
```

También podemos monitorizar un puerto o interfaz en específico del switch:

```
define service{
     use                        generic-service ; Inherit values from a template
     host_name                  Switch_1
     service_description        Port 12 Link Status
     check_command      check_snmp!-C public -o ifOperStatus.1 -r 1 -m RFC1213-MIB
    }
```

(Hay que añadir la cantidad de servicios a ser monitorizados por cada host que hayamos añadido, todo eso dentro de este archivo).

Listo, nos aseguramos que no hayan errores y reiniciamos nagios

```
#/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
#/etc/init.d/nagios restart
```

Ahora, lo que tenemos es un Servidor Nagios de monitorización un poco básico, el cual nos da información medianamente básica, leyendo el Manual de Nagios (**www.nagios.org**) podemos hacer o aumentar más cosas.

Pero, hay una parte que no se ve bien, que es la del mapa, ya que en todos los dispositivos que pongamos, nos saldrá un horrible signo de interrogación. Pero eso lo podemos cambiar.

Bueno, esto es fácil, básicamente entramos a **www.nagiosexchange.org** , buscamos los packs o el pack de imágenes que más nos guste, por ejemplo a mi me gusta nagios, ya que puedo personalizar incluso la imagen de cada una de mis máquinas Linux, según su distribución, impresoras, wi-fi, etc.

Lo bajo, lo desempaqueto (porque es un **.tar.gz**) y copio todas las imágenes que quiero en:

```
/usr/local/nagios/share/images/logos
```

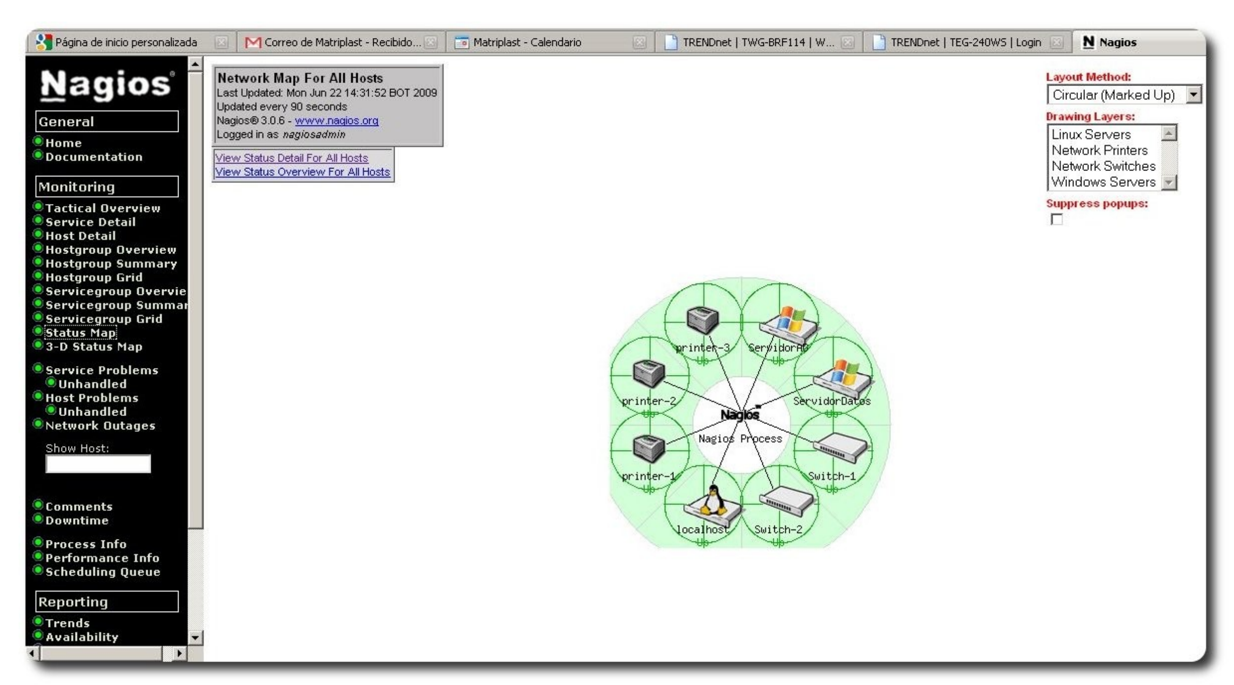


Diagrama 4: Mapa del Estado de los dispositivos en red

Fuente: Elaboración Propia

Cada icono suele venir en formatos **.png**, **.jpg**, **.gif** y **.gd2**, ésto porque se usan diferentes formatos para diferentes lugares.

Ahora enlazamos la imagen que queremos con cada uno de nuestros hosts, ésto en donde esté defino el host, por ejemplo

```
#vi /usr/local/nagios/etc/objects/localhost.cfg
```

y ponemos:

```
   define host{
       use              linux-server  ; Name of host template to use
       host_name        localhost
       alias            localhost
       icon_image       ubuntu.gif
       statusmap_image  ubuntu.gd2
       address          127.0.0.1
     }
```

Ahora podré tener el Logo de Ubuntu en mi Servidor Ubuntu, o a Tux para poder diferenciar mis máquinas Linux de las de Windows.

Y así en cada uno de los hosts que estemos monitorizando, saldrá su logo, en el mapa así como en la lista.

También es importante mostrar cuan gráfica es la administración de Nagios vía web, donde aparte de tener el mapa de los dispositivos, también nos muestra pequeños resúmenes de status, para saber de forma rápida y concisa si se tiene algún problema con algún dispositivo:

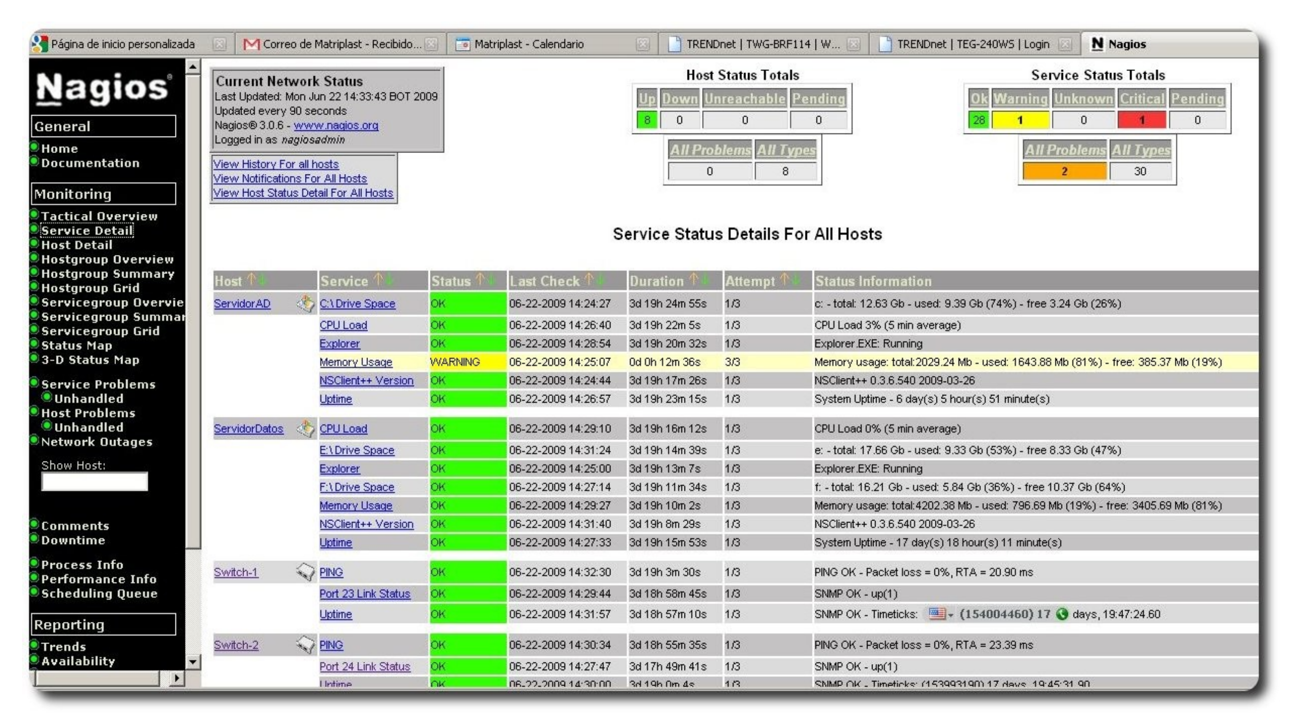


Diagrama 5: Resumen de los hostgroups

Fuente: Elaboración Propia

Si queremos también podemos acceder a esta información de forma más detallada haciendo clic en "Service Details", el cual nos dará una lista de todos los servicios en todos los dispositivos:

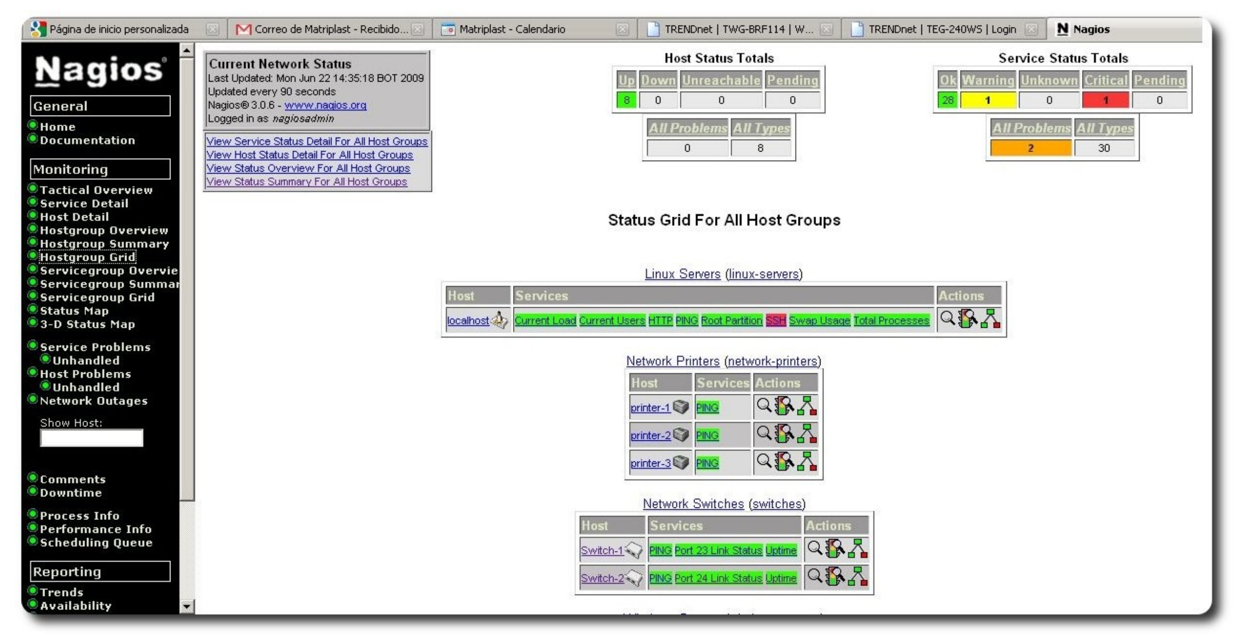


Diagrama 6: Detalle de los Servicios de todos los hosts

Fuente: Elaboración Propia

Conclusiones

Para concluir el Artículo, se debe recalcar que Nagios es una herramienta de monitoreo, ésto quiere decir que nos ayuda a ver si hay fallas o problemas, pero ya depende de nosotros el ver la causa y el como solucionar dicho problema.

Una gran ventaja es que al ser modular, nosotros con el tiempo podemos ir añadiendo más equipos a monitorizar (según como se vayan añadiendo estos equipos a la red).

Finalmente se debe notar que al ser software libre, éste es completamente accesible y si fuera necesario nosotros mismos podemos hacerle mejoras según nuestras necesidades y compartirlas con los demás.

Referencias

[1] Nagios 3.x Documentación Oficial http://nagios.org/docs/

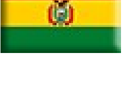
[2] Wiki de la comunidad Nagios
http://community.nagios.org/wiki/index.php/Main_Page

[3] Nagios Exchange http://www.nagiosexchange.org

Autor



Ivan Gutierrez Agramont
Magister en Telecomunicaciones y Telemática
Ingeniero de Sistemas
Miembro de la Comunidad de Software Libre Bolivia
Entusiasta Linux (Ubuntu y sus derivados, Trixbox, Kamilio y otros)
ivan.guag@gmail.com



Proyectos Latinos Finalistas del Community Choice Awards



Apoya al proyecto de tu preferencia



Elastix Finalista por Tercer Año en los CCA de Source Forge

Elastix es un proyecto de Software Libre desarrollado en Ecuador por la compañía Palosanto Solutions. Esta herramienta permite crear servidores de Comunicaciones Unificadas basados en software libre integrando los servicios: VoIP, fax, correo electrónico y chat.

Para una extensión telefónica que utiliza protocolos Voz/IP le es prácticamente lo mismo estar a pocos metros de la central dentro de una red LAN o a miles de kilómetros a través del Internet. Esto trae beneficios interesantes para los usuarios de Elastix como poder contestar la extensión telefónica de la oficina desde cualquier parte del mundo. La integración de servicios es otra ventaja útil, por ejemplo, un usuario podrá recibir en su correo electrónico los buzones de voz o faxes en formato pdf.

El proyecto Elastix tiene muy buena documentación y prueba de esto son los libros “Comunicaciones Unificadas con Elastix Vol. 1&2” bajo licencia GNU FDL en español y el libro “Elastix Without Tears” bajo licencia CC en inglés. Estos libros están disponibles en el área de descarga del sitio web de Elastix.

En el año 2006 Elastix fue liberado y publicado en el portal de Source Forge y desde entonces ha tenido mucho éxito. En este momento tiene cerca de 500 000 descargas y ha sido finalista tres años consecutivos a los premios de la Comunidad de Source Forge. Esta año Elastix es finalista en la categoría “Best Project for the Enterprise” entre 4875 proyectos. ¡Vota por Elastix y apoya al software libre de América Latina.!

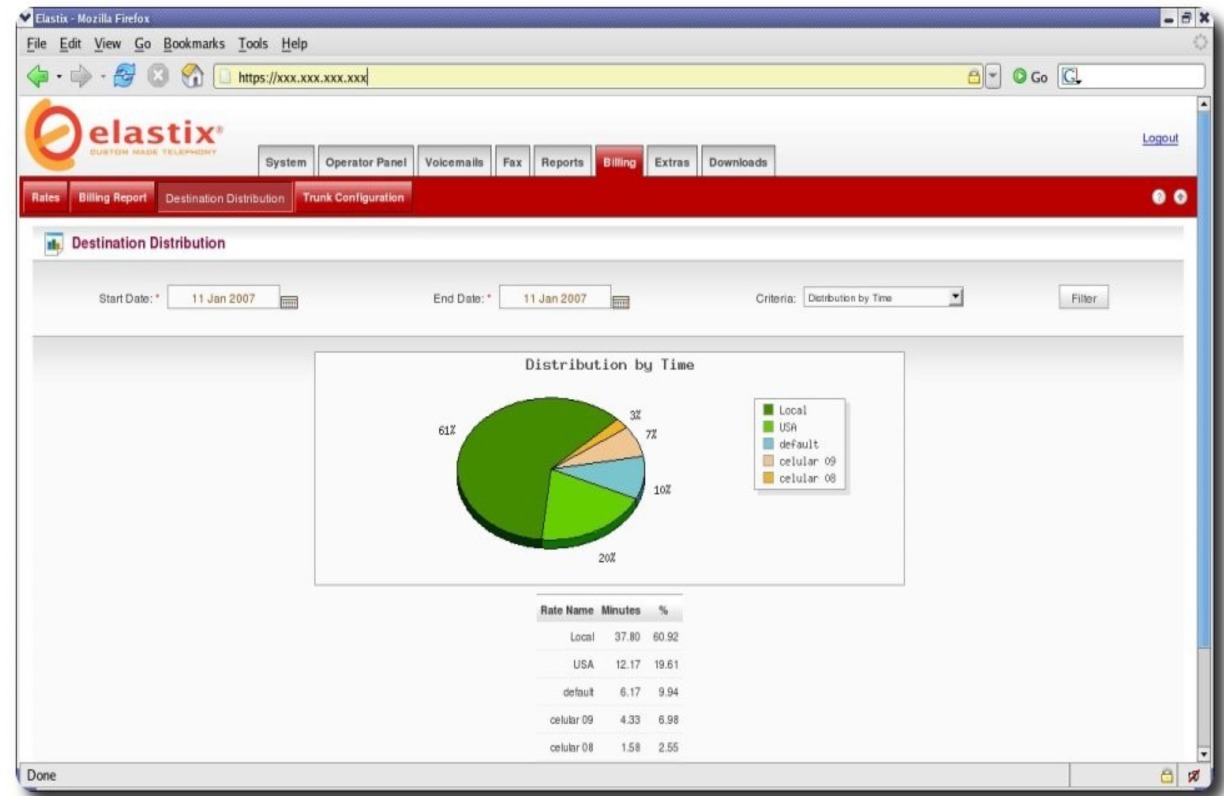
Para votar por Elastix: http://sourceforge.net/community/cca09/vote/?f=425

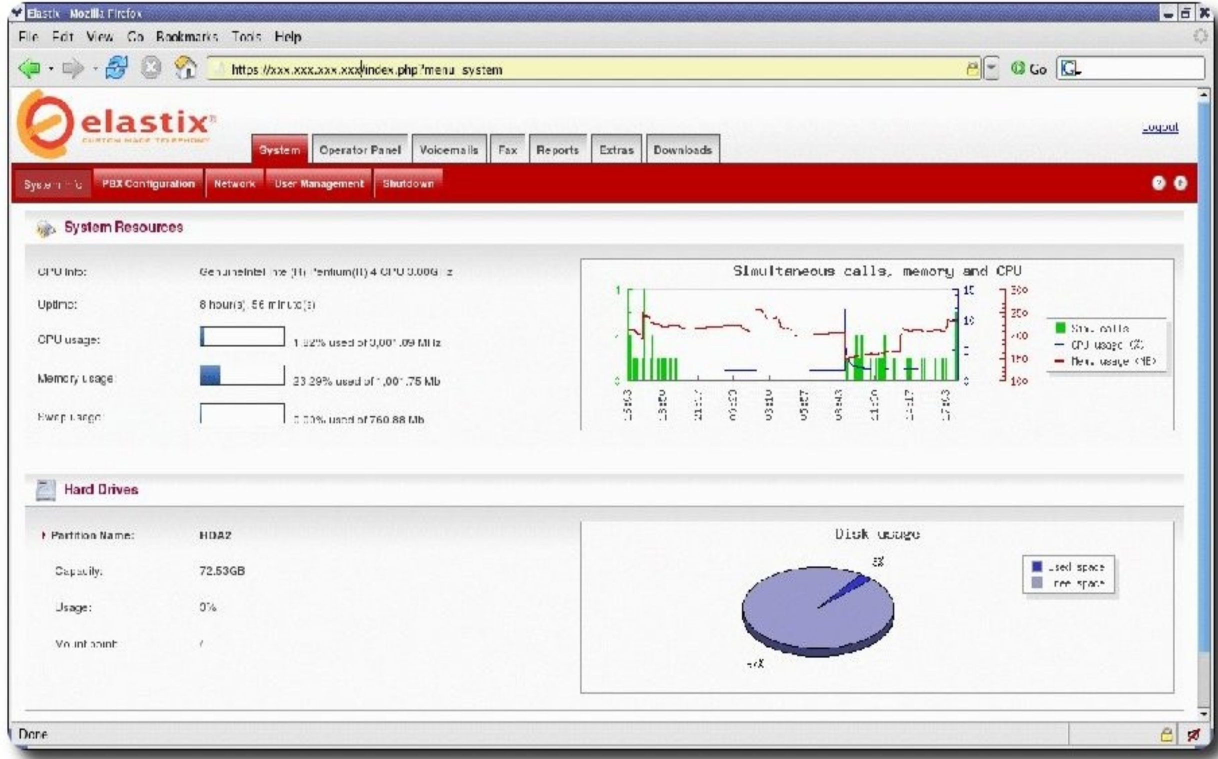
Sitio web de Elastix: http://www.elastix.org





Finalista


Elastix - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
https://xxx.xxx.xxx.xxx
elastix
System
Operator Panel
Voicemails
Fax
Reports
Billing
Extras
Downloads
Logout
Rates
Billing Report
Destination Distribution
Trunk Configuration
Destination Distribution
Start Date: * 11 Jan 2007
End Date: * 11 Jan 2007
Criteria: Distribution by Time
Filter
Distribution by Time
61%
3%
7%
10%
20%
Local
USA
default
celular 09
celular 08
Rate Name Minutes %
Local 37.80 60.92
USA 12.17 19.61
default 6.17 9.94
celular 09 4.33 6.98
celular 08 1.58 2.55
Done


elastix
System
Operator Panel
Voicemails
Fax
Reports
Extras
Downloads
PBX Configuration
Network
User Management
Shutdown
System Resources
CPU Info:
Uptime:
CPU usage:
Memory usage:
Simultaneous calls, memory and CPU
Hard Drives
Partition Name:
HDA2
Capacity:
72.53GB
Usage:
0%
Disk usage
Done



ProcccessMaker un software de gestión de procesos

ProcessMaker es el primer software de gestión de procesos y flujos de trabajo verdaderamente open source orientado a PyMEs y unidades de negocio. ProcessMaker incluye herramientas de uso fácil que posibilitan la gestión eficiente y efectiva de procesos operacionales a través de sistemas incluyendo finanzas, recursos humanos y operaciones. Esto permite que gerentes de negocio y expertos en procesos que no tengan experiencia en programación puedan modelar y automatizar procesos cotidianos, incrementando transparencia y reduciendo radicalmente el papeleo.

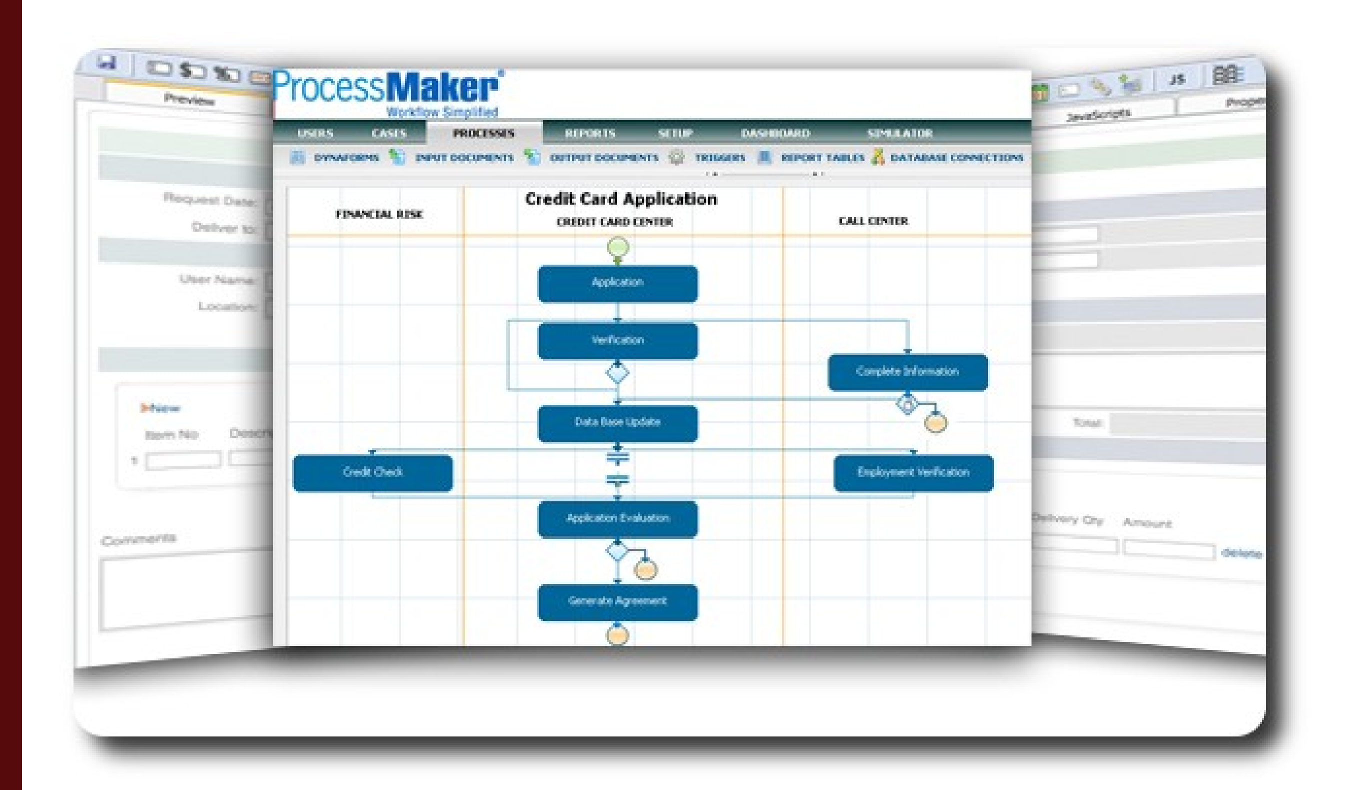
ProcessMaker es desarrollado enteramente con talento Boliviano. Hace un año y tres meses decidimos cambiar de modelo de Negocio y nos fuimos por el 'Open Source' liberando el código y dando paso a la creación de una Comunidad alrededor del producto ProcessMaker. Tener un producto Open Source tiene varias ventajas desde el hecho que la comunidad nos ayuda a mejorar el producto hasta la oportunidad de participar en eventos junto a los grandes del Open Source como Openbravo, KnowledgeTree, SugarCRM, etc.

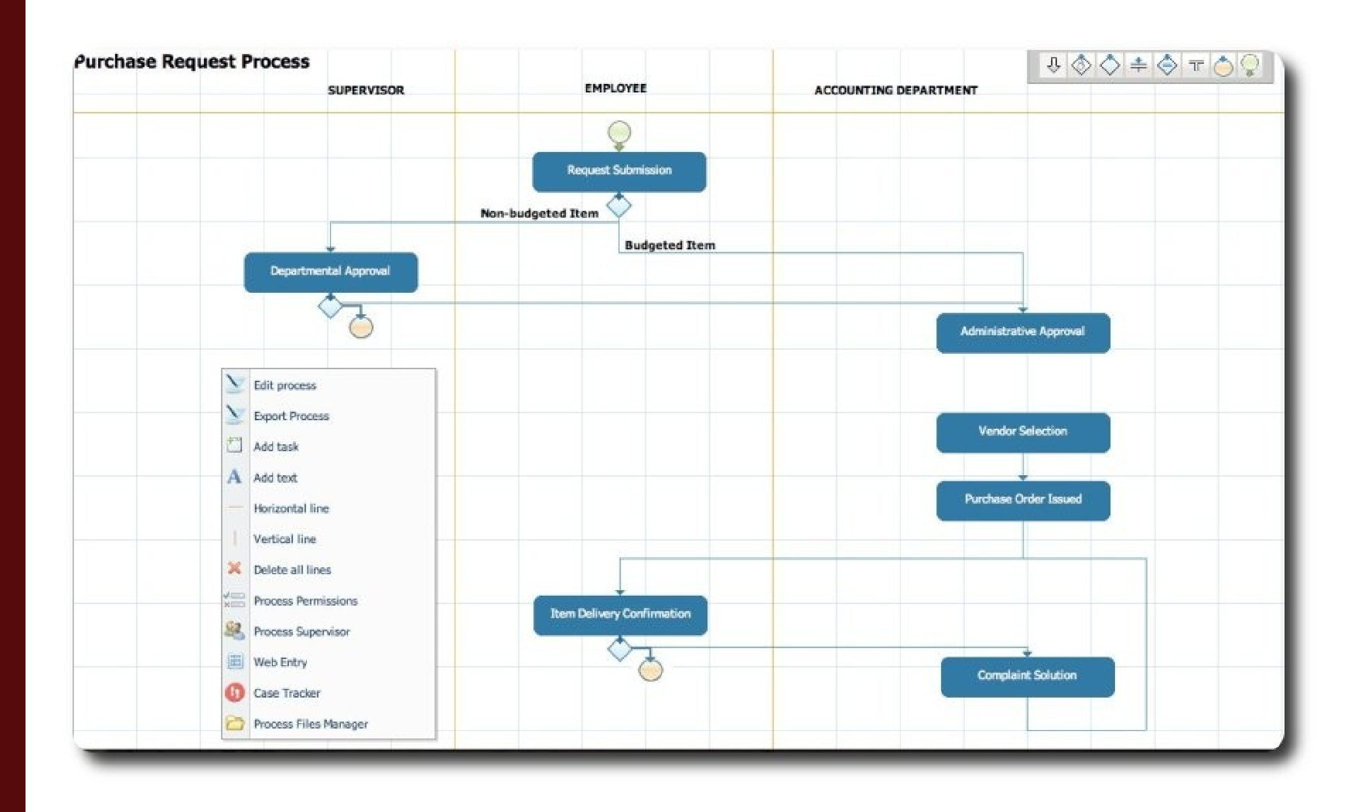
Estar nominados entre los 10 mejores de SourceForge nos llena de gran satisfacción y nos motiva a seguir adelante.



Community Choice Awards

Finalista





Medical : El sistema de Información de Salud Universal

El objetivo del proyecto Medical es mejorar la calidad de vida de las poblaciones, especialmente de los países en vía de desarrollo. La información epidemiológica suministrada por Medical permitirá a las ONGs y entidades de salud mejorar las campañas sanitarias y detectar tempranamente brotes de epidemias (VIH/SIDA, Chagas, Malaria, ...). Medical hace hincapié en la medicina familiar, identificando la relación entre condiciones socio económicas de las familias con el estado de salud de las mismas.

Medical es un sistema de Expediente Médico Electrónico (EMR), un Sistema de Gestión Hospitalaria (HIS) y un Sistema de Información de Salud.

Es universal porque es abierto, en el más amplio sentido de la palabra: Es código abierto (open source); lo pueden utilizar instituciones y doctores en todo el mundo, independientemente de su status económico; utiliza estándares abiertos de la industria (ICD-10 / CIE-10) y cuenta con un equipo de doctores, bioquímicos, sociólogos e informáticos de distintos países.

La Funcionalidad principal abarca :

- ✔ Administración de Paciente (generación de la HC, seguimiento,citas, derivaciones, reingresos, etc)
- ✔ Fuerte interés en estatus socio económico (condiciones de viviendas, educación, abuso de drogas, trabajo infantil...)
- ✔ Administrador del profesional médico (calendario de consultorio, cirugías, interconsultas...)
- ✔ Gestión de de estudios complementarios (estudios de laboratorio, imagenología)
- ✔ Vademécum
- ✔ Gestión de stock y abastecimiento (incluye medicamentos e insumos médicos en el caso de instituciones)
- ✔ Estadísticas sobre poblaciones / patologías / grupos etarios ...
- ✔ Administración Financiera (facturación, ...)

Open Source. Licenciado bajo GPL v3

Página del proyecto : http//medical.sourceforge.net

File User Form Options Plugins Shortcuts Help

New Save Delete Back Forward List Form Calendar Graph Print Action Archivos adjuntos Menu Reload Close

Menu | Nuevo Paciente

Información Principal

Paciente : Esperanza | Sexo : Femenino

Fecha de nacimiento : 07/07/2002 | Edad del Paciente : 7 0 1

Grupo étnico : | Estado civil : Soltero/a

Grupo Sanguineo : A | Rh : +

Estilo de vida, Socio-económicos, Vacunas, Historia Familiar, Alergias, Medicamentos, Cirugías, Genético, Crónico, Sexualidad / ETS, Infancia, Psiquiátricas, Ginecológico, Evaluaciones, Notas

Principal

Socio-económicos : Baja

Nivel de Educación : Ninguno

APGAR Familiar

Ayuda familiar : Moderado | Discusión de problemas en el ámbito familiar : Moderado

Toma de decisiones familiar : Ninguno | Tiempo compartido en la familia : Moderado

Afecto familiar : Mucho

Otros problemas familiares

Familia de padre / madre soltero : | Violencia Doméstica :

Niños que trabajan : | Embarazo adolescente :

Abuso sexual : | Adicción a las drogas :

Abandono Escolar : | Ha estado en prisión :

Está actualmente en prisión : | Familiar en prisión :

Trabajo

Profesión : | Trabaja en casa :

Horas fuera de casa : 0

Infraestructura

Condiciones de la vivienda : Condiciones sanitarias prec

Sistema cloacal : | Agua potable :

Recolección de basura : | Suministro Eléctrico :

Gas : | Teléfono :

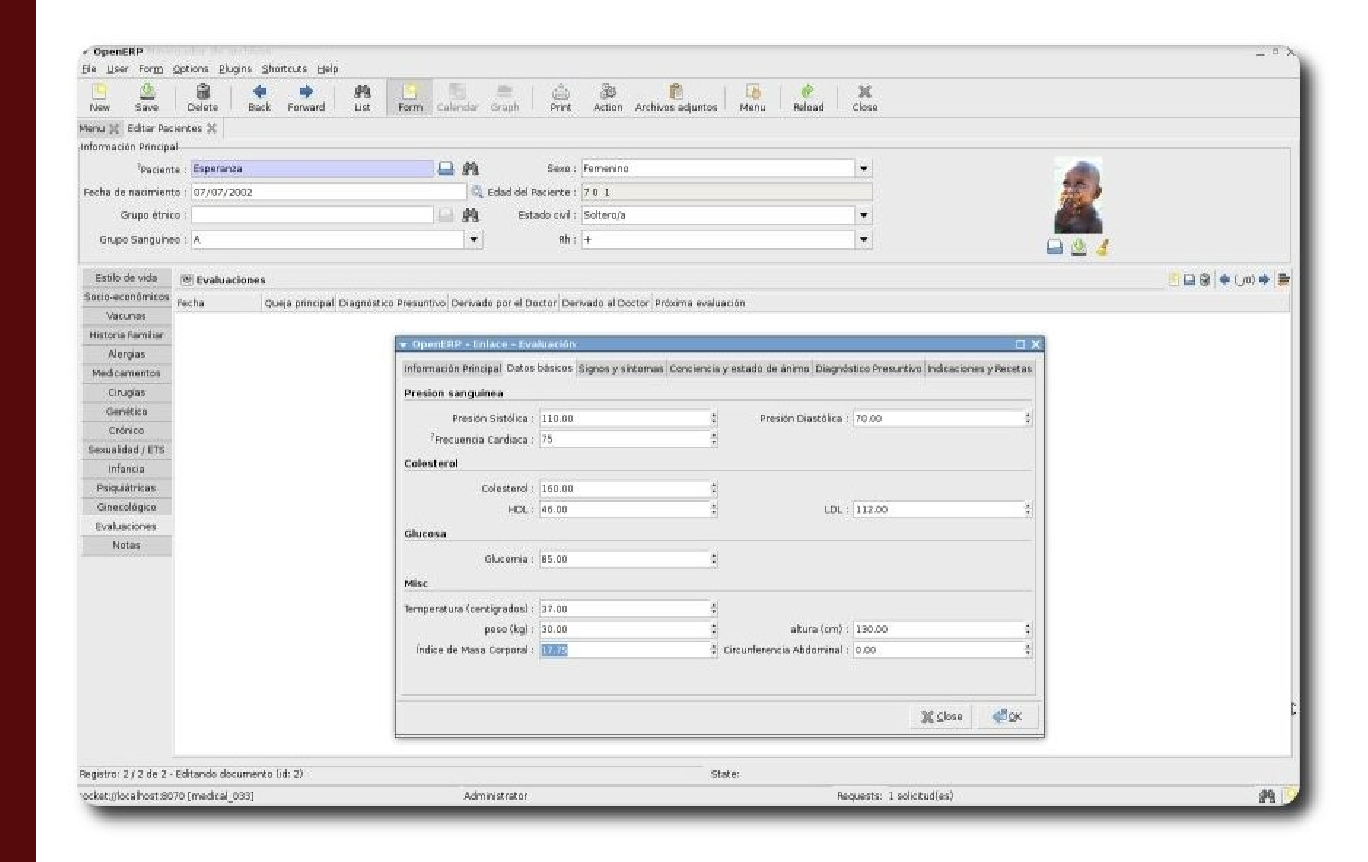
Televisión : | Internet :

Información Adicional

Area Hostil :

Ningún registro seleccionado | State: Documento guardado.

socket://localhost:8070 [medical_033] | Administrator | Requests: 1 solicitud(es)




Willay news
Información Actual

Willay news

Auto Ecológico y Open source

Riversimple compañía británica dedicada a la construcción de automóviles de buen rendimiento energético, lanzó en Londres un nuevo modelo denominado **Riversimple Urban** un auto ecológico liviano a base de hidrógeno y además es open source.



En 1999 el ex piloto de carreras y fundador de Riversimple Hugo Spowers tuvo una magnífica idea, les hablamos de lo que hoy es el denominado Riversimple Urban, un auto ecológico compuesto por cuatro motores impulsados por una pila de combustible de 6kw y además presenta los ultracondensadores que almacenan grandes cantidades de carga eléctrica. Riversimple alcanza una velocidad máxima de 80 km/h y puede viajar hasta 390 kilómetros con un estanque de un kilogramo de hidrógeno.

El Riversimple urban es open source, lo que significa que la compañía entregará los planos de ingeniería del automóvil a 40 Fires Fundation, una organización dedicada al desarrollo de coches con alta eficiencia energética, para que esta organización pueda disponer los diseños al alcance de todos quienes estén interesados en mejorar el modelo.

Algunos creen que el hidrógeno no es el combustible del futuro, ya que es muy caro de conseguirlo, de todas maneras el Riversimple urban es un aporte beneficioso y amigable al medio ambiente. Probablemente el automóvil del futuro partirá del Riversimple urban y de seguro tendremos noticias cuando menos lo imaginemos.



Congreso Gran Canaria Desktop Summit 2009

Las islas canarias será la sede del Congreso Gran Canaria Desktop Summit 2009.



Este año presenta una agenda interesante en cuanto a congresos y ferias importantes a nivel mundial, haciendo un repaso sobre el primer semestre: en febrero se llevó a cabo el Mobile World Congress (Barcelona del 16 al 19 de Febrero), en marzo se llevó a cabo la mayor feria mundial de Tecnologías de información y telecomunicación CEBIT (Hannover Alemania desde el 3 al 8 marzo), en abril celebrando los 20 años de la web El World Wide Web 2009 (Madrid del 20 al 24 de abril), en junio se llevó a cabo en Los Ángeles el E3 2009 (Electronic Entertainment Expo 2009) la mayor feria de juegos del mundo.

Empezando el segundo semestre de este año se tiene preparada el Congreso Gran Canaria Desktop Summit 2009 del 3 al 11 de julio en el Auditorio Alfredo Kraus, la Universidad de Las Palmas de Gran Canarias y algunas otras instituciones que hacen posible este evento, que reunirá a las principales comunidades de software libre a nivel mundial, con la participación de las fundaciones (KDE y GNOME) para escritorio y portátiles. Este congreso reunirá a los principales gurús del software libre y tendrá como objetivo trabajar en las nuevas versiones de ambos escritorios que por primera vez están juntos en un evento de esta naturaleza; sin duda este congreso tiene una gran expectativa, para todos aquellos fieles seguidores del software libre.

Autor

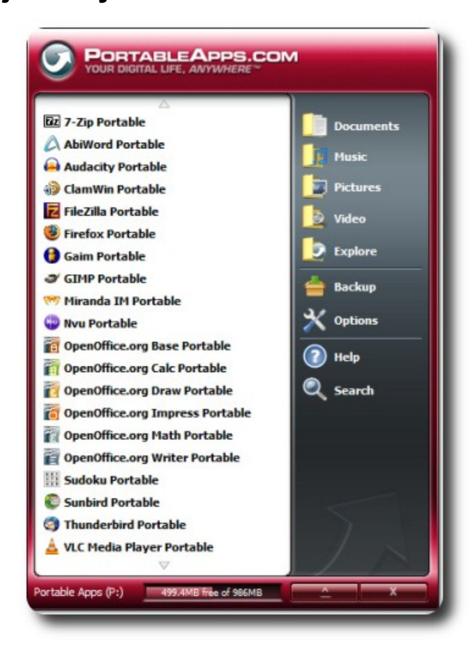


Ivonne Karina Menacho Mollo
Titulada de la carrera de Ingeniería de Sistemas e Informática (F.N.I.)
Conductora programa radial “**Doble Clic**”
ikmenacho@gmail.com

Willay news

PortableApp.com

“**Tu vida digital donde quieras**” es sinónimo de PortableApp.com. Esta increíble plataforma portátil, está nominada en Ocho categorías en Community Choice Awards de SourceForge.net, incluyendo “**Mejor Proyecto**” .



Detengámonos a pensar, tener siempre a mano y poder acceder a un navegador Web, un gestor de correo, mensajería instantánea, reproductor de música, un paquete de programas para ofimática, lector de documentos PDF, administrador de contraseñas, agenda, antivirus y juegos, desde cualquier dispositivo de almacenamiento portátil: PenDrive, iPod/MP3, etc.



PortableApp.com cuenta con todo lo mencionado, que le valió sus ocho nominaciones, y más. De las aplicaciones que puedan tentar están: Un editor Web “Nvu y KompoZer portátil” similar a Dreamweaver cuenta con un administrador de sitios FTP, edición CSS, etc. Un servidor portátil XAMPP que cuenta con Apache, mySQL, PHP. todo pre-configurado. Notepad ++ listo par usar. Stellarium programa de astronomía, muestra un cielo realista en 3D, planetas con impresionante detalle. Algo que impresione puede ser, el de ejecutar una Mac OS Classic 7 portátil con Mac-os-Stick



Su funcionamiento es simple: Solo conectas tu PenDrive a un ordenador, y accedes a tus programas e información, tal como lo harías desde tu propia computadora, terminas de trabajar, desconectas tu PenDrive, y absolutamente nada de tu información se quedara en esa PC.

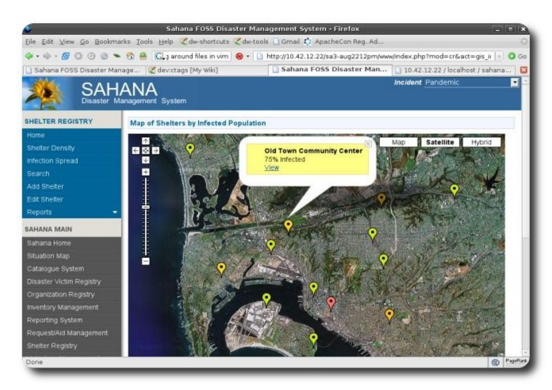
El paquete se puede descargar en 3 modalidades, que ocuparan 1.3MB, 150 MB y 350 MB, también se pueden realizar descargas por separado de cualquier aplicación portátil que uno guste.

Aunque solo se pueda usar en cualquier ordenador con Windows, lo mejor de este gran paquete de aplicaciones portátiles: es el de ser totalmente LIBRE toda la plataforma es de código abierto (incluyendo todos los programas que contiene), realmente se trata de una Suit de Aplicaciones portátiles que puedes llevar en el bolsillo, para servirte en cualquier momento.

Sistema de Gestión de Desastres: SAHANA

En cada uno de nuestros países hemos sufrido algún desastre natural, muchos de nosotros quizá hasta ahora sentimos las secuelas de aquellas inesperadas y terribles: inundaciones, incendios, terremotos, sequías, tormentas, etc. que sin duda alguna, ocasionaron pérdidas tanto humanas como materiales, a centenares de familias.

SAHANA un extraordinario Sistema de Gestión de Desastres hace uso de las TICs proveyendo un conjunto integrado de aplicaciones de manejo de desastres modulares, basados en Web, que brinda soluciones a problemas humanitarios de gran escala; durante, mientras y después de un desastre.



Cuenta con seis módulos principales:

- Registro de personas extraviadas: mantiene la información de personas extraviadas y de las que la están buscando.
- Registro de organizaciones de ayuda y servicio, ayuda a determinar "**quien hace que y donde**."
- Registro de albergue: con cuantos albergues se cuenta y la capacidad de los mismos.
- Sistema de petición/ayuda: se coordina los requerimientos de un determinado sector o campamento con las organizaciones de ayuda y gobierno.
- Sistema de coordinación de voluntarios: se tiene un seguimiento de todos sus voluntarios, disponibilidad y proyectos para una ubicación efectiva.
- Sistema de análisis de situación: se tiene las características de determinada situación, se puede colocar una imagen o información sobre un mapa.

Este magnifico proyecto cuenta también con una comunidad de voluntarios apasionados para su desarrollo, debido a que todo el sistema es LIBRE y gratuito, además de estar desarrollado en tecnologías como Apache, MySQL, PHP/Perl.



SAHANA tiene un gran significado el cual es: "**alivio**" y quizá con una pequeña ayuda que cualquiera de nosotros pueda dar, demos eso mismo: ALIVIO, recordando que, los DESASTRES llegado el momento no piensan en credo, posición social o raza.

Autor



Marcia Estrella Velasquez Aguilar
Egresada de la carrera de Ingeniería de Sistemas (F.N.I.)
mevaguerreradelaluz@gmail.com

Willay news

Cargador MicroUSB

La Comisión Europea, junto a algunos fabricantes de teléfonos móviles, han llegado a un acuerdo para utilizar el puerto MicroUSB como conector de corriente en los teléfonos móviles.

Esta estandarización traerá numerosas ventajas tanto a los usuarios como a los fabricantes. Por ejemplo, ya sólo necesitaremos tener un único cargador para poder alimentar todos y cada uno de los móviles. Y lo más probable es que conforme avance el tiempo, algún fabricante lance bases con varias tomas MicroUSB para poder cargar varios terminales de forma simultánea.



Por el lado de los fabricantes esta medida hará que cesen en la investigación de los cargadores, las patentes y el desarrollo de estos accesorios. Además también podrán dejar de incluirlos junto con los terminales, de forma que podrán ahorrar el costo en su fabricación para de esta forma reducir el precio final del dispositivo.

Ahora ya solo queda que este tipo de estandarización de cargadores se traslade a otros dispositivos, como por ejemplo portátiles, reproductores multimedia o GPSs, entre otros

Más allá de la incomodidad que esto supone para el usuario, hay una cuestión ecológica de fondo que preocupa mucho más: ¿qué hacemos con el cargador cuando cambiamos de teléfono? Como es tan específico, a no ser que se compre un modelo muy similar, sólo queda el recurso de llevarlo a un punto de reciclaje, ya que no debe tirarse a la basura. De esta suerte, la incompatibilidad entre cargadores de móviles genera al año entre 51.000 y 82.000 toneladas de basura tecnológica. Esta basura se produce porque cada año se renuevan entre el 50% y el 80% de los 1.200 millones de móviles vendidos.



Esperemos que este tipo de iniciativa triga consigo muchos beneficios.

Porta retratos Kodak

El nuevo Kodak EasyShare S730 nos brinda la oportunidad de tener un porta retratos dinámico, el cual pueda contener las fotos que queramos en un determinado instante o la combinación de las mismas.

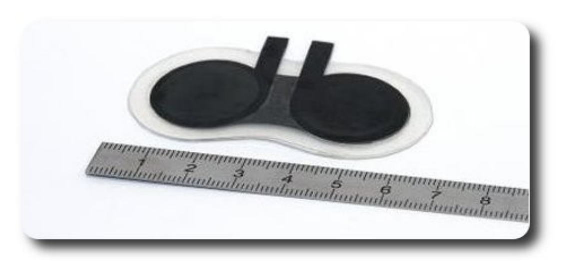


El marco Kodak S730 dispone de una pantalla de 7 pulgadas con resolución de 800×480 píxeles y memoria interna de 1 GB, además de puerto USB y lector de tarjetas de memoria. Con esa pantalla y resolución, Kodak también ha optado por que su nuevo marco de fotos pueda reproducir vídeo en bastantes formatos, entre ellos AVI, MPEG4 y MOV. La reproducción del contenido lo podemos controlar de forma táctil, pero no en la pantalla sino en el marco.



Baterías impresas en cualquier lugar

Las baterías siguen siendo una de las partes que más limitan nuestros dispositivos. Su capacidad no ha crecido tanto como el consumo de estos y, si queremos aumentar la autonomía de momento no queda más remedio que hacerlas crecer de tamaño. Unos científicos alemanes están trabajando en un modelo ultradelgado.



Estas nuevas baterías tienen un grosor inferior a un milímetro y pesan menos de un gramo, permitiendo imprimirlas sobre la superficie donde queremos instalarlas.

Producen un voltaje de 1.5 voltios, pero pueden combinarse para conseguir hasta 6 voltios sin hacer crecer demasiado su grosor. Eso sí, como no podía ser de otro modo tienen una desventaja, el cátodo y el ánodo se disipan en el proceso de generar energía, por lo que tienen una vida limitada.

De todos modos, siguen teniendo una gran camntidad de aplicaciones y sus creadores afirman que están listos para producirlas de forma comercial hacia finales de este año.

Autor



Jenny Saavedra López
Diseño y Edición Revista Atix
jennysaavedra@gmail.com



Una Revista mas
alla de lo evidente

Comics


Me gustaría controlar por ordenador mis gastos de casa ¿Cómo lo harías?


Yo prepararía en **EXCEL** una base de datos, después lo encapsularía en **BASIC** para hacer un script en **JAVA** y lo editaría todo en ficheros navegables **HTML**


Creo que continuaré con mi libreta de anillas de "última generación"
Softonic-JSánchez 2005


En mi empresa todo el mundo tiene ya su propio **blog**


¡pues vaya cosa!, en la mía también...precisamente llevo el mío aquí...


mira que **bloc** más chulo, con sus tapas duras y separadores de colores!



Conociendo lo Nuestro

Santa Cruz

Plaza principal Santa Cruz de la Sierra



Vista aérea de Santa Cruz



Catedral de Santa Cruz



Misiones Jesuíticas Santa Cruz



Parque Acuático Aqualand



Paisaje a las fueras de Santa Cruz

Libres para pensar, libres para decidir, libres para crear

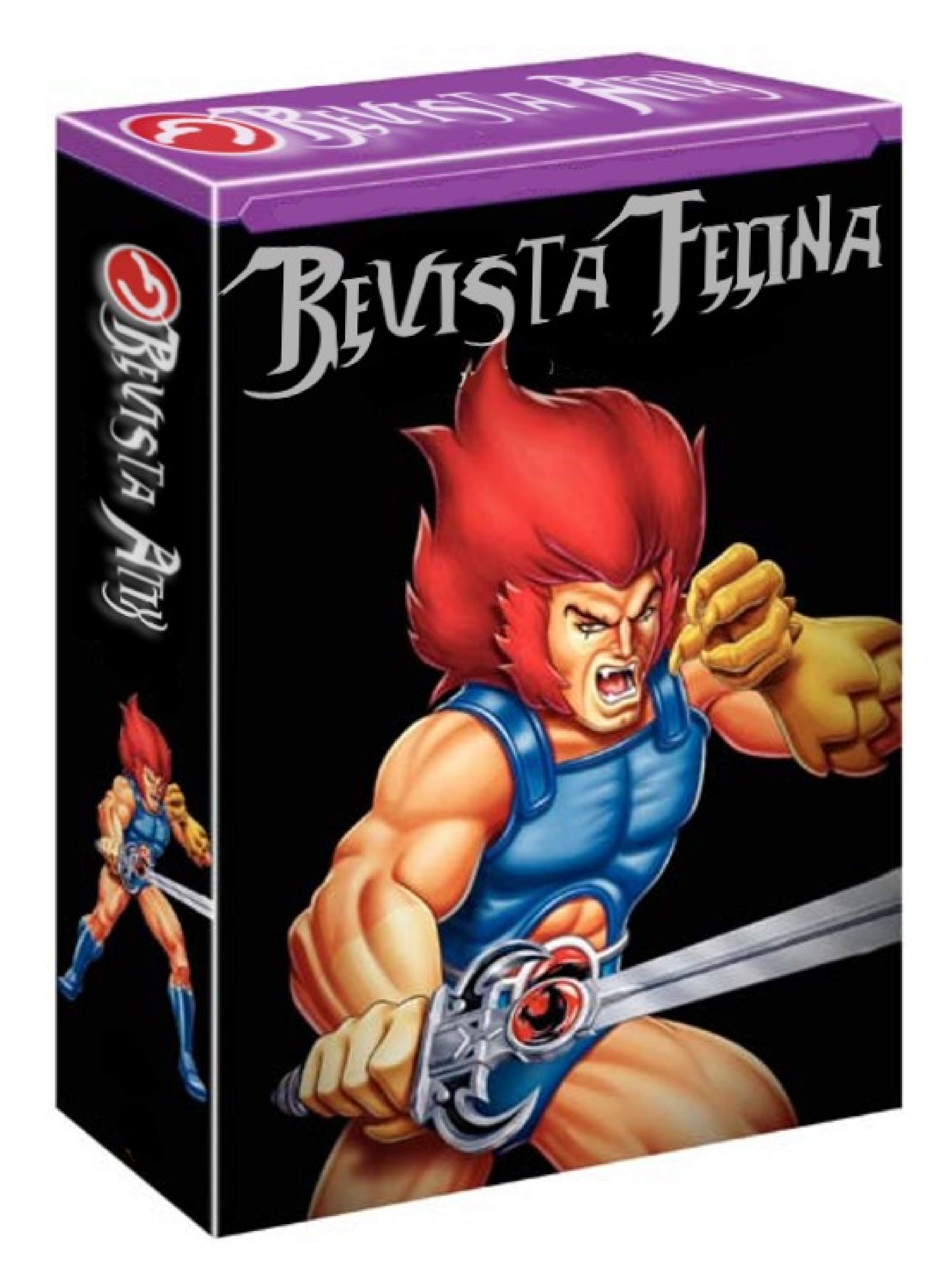


Te ofrecemos este espacio para mostrar tu Creatividad

ATIX

La Fuerza del Software Libre

Envíanos tus diseños y creaciones para publicarlos


REVISTA FELINA

Consultas

Contacto

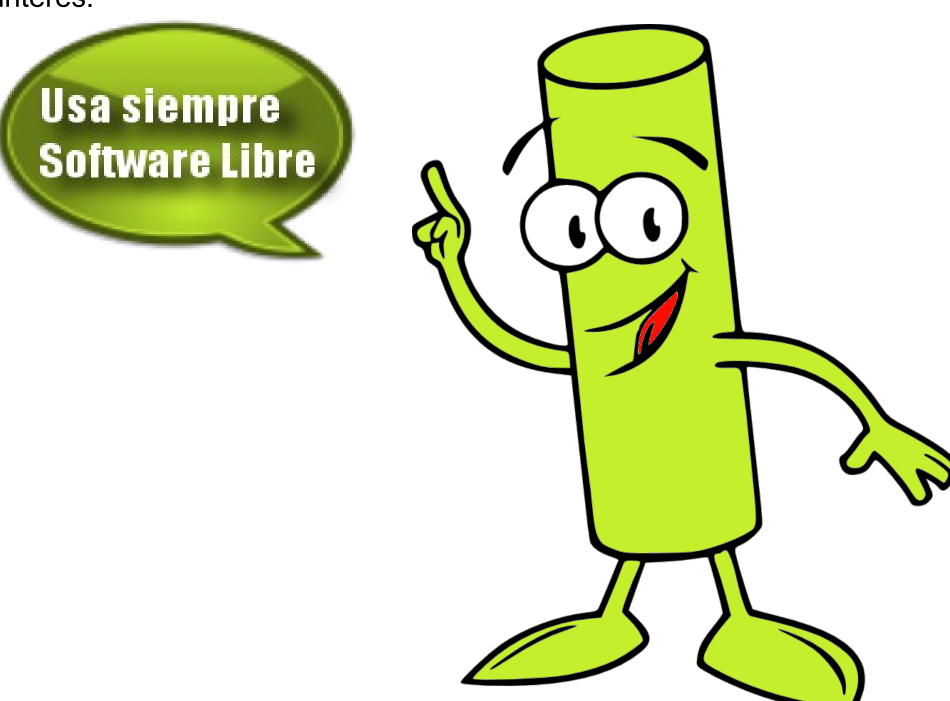
Para solicitar cualquier información, puedes contactar a:

- Esteban Saavedra López (jesaavedra@opentelematics.org)
- Jenny Saavedra (jennysaavedra@gmail.com)

Publicación

Te invitamos a ser parte de la **Revista ATIX**. La forma de participar puede ser enviándonos:

- Artículos referidos a áreas como:
  - Instalación y personalización de Aplicaciones
  - Scripting
  - Diseño gráfico
  - Programación y desarrollo de aplicaciones
  - Administración de servidores
  - Seguridad
  - y cualquier tema enmarcado dentro del uso de Software Libre
- Trucos y recetas.
- Noticias.
- Comics.
- Links de interés.



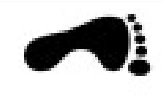
Bitacora

Empezamos a registrar nuestra historia

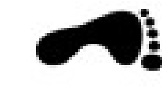
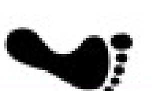




Marcamos Huella





http://atix.opentelematics.org